Ph. D DISSERTATION

Document Image and Scene Text Rectification via Text and Feature based Optimization

텍스트와 특징점 기반의 목적함수 최적화를 이용한 문서와 텍스트 평활화 기법

BY

BEOM SU KIM

AUGUST 2014

DEPARTMENT OF
ELECTRICAL AND COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Document Image and Scene Text Rectification via Text and Feature based Optimization

텍스트와 특징점 기반의 목적함수 최적화를 이용한
문서와 텍스트 평활화 기법

지도교수 조 남 익
이 논문을 공학박사 학위논문으로 제출함
2014 년 6 월

서울대학교 대학원
전기·컴퓨터 공학부
김 범 수

김범수의 공학박사 학위논문을 인준함
2014 년 6 월

위 원 장 : 이 상 욱

부위원장 : 조 남 익

위 원 : 최 진 영

위 원 : 이 정 우

위 원 : 김 창 수

ABSTRACT

There are many techniques and applications that detect and recognize text information in the images, e.g., document retrieval using the camera-captured document image, book reader for visually impaired, and augmented reality based on text recognition. In these applications, the planar surfaces which contain the text are often distorted in the captured image due to the perspective view (e.g., road signs), curvature (e.g., unfolded books), and wrinkles (e.g., old documents). Specifically, recovering the original document texture by removing these distortions from the camera-captured document images is called the document rectification. In this dissertation, new text surface rectification algorithms are proposed, for improving text recognition accuracy and visual quality. The proposed methods are categorized into 3 types depending on the types of the input. The contributions of the proposed methods can be summarized as follows.

In the first rectification algorithm, the dense text-lines in the documents are employed to rectify the images. Unlike the conventional approaches, the proposed method does not directly use the text-line. Instead, the proposed method use the discrete representation of text-lines and text-blocks which are the sets of connected components. Also, the geometric distortion caused by page curl and perspective view are modeled as generalized cylindrical surfaces and camera rotation respec-

tively. With these distortion model and discrete representation of the features, a cost function whose minimization yields parameters of the distortion model is developed. In the cost function, the properties of the pages such as text-block alignment, line-spacing, and the straightness of text-lines are encoded. By describing the text features using the sets of discrete points, the cost function can be easily defined and well solved by Levenberg-Marquadt algorithm. Experiments show that the proposed method works well for the various layouts and curved surfaces, and compares favorably with the conventional methods on the standard dataset.

The second algorithm is a unified framework to rectify and stitch multiple document images using visual feature points instead of text lines. This is similar to the method employed in general image stitching algorithm. However, the general image stitching algorithm usually assumes fixed center of camera, which is not taken for granted in capturing the document. To deal with the camera motion between images, a new parametric family of motion model is proposed in this dissertation. Besides, to remove the ambiguity in the reference plane, a new cost function is developed to impose the constraints on the reference plane. This enables the estimation of physically correct reference plane without prior knowledge. The estimated reference plane can also be used to rectify the stitching result. Furthermore, the proposed method can be applied to any other planar object such as building facades or mural paintings as well as the camera-captured document image since it employs the general features.

The third rectification method is based on scene text detection algorithm, which is independent from the language model. The conventional methods assume that a character consists of a single connected component (CC) like English alphabet. However, this assumption is brittle in the Asian characters such as Korean, Chinese,

and Japanese, where a single character consists of several CCs. Therefore, it is difficult to divide CCs into text lines without language model. To alleviate this problem, the proposed method clusters the candidate regions based on the similarity measure considering inter-character relation. The adjacency measure is trained on the data set labeled with the bounding box of text region. Non-text regions that remain after clustering are filtered out in text/non-text classification step. Final text regions are merged or divided into each text line considering the orientation and location. The detected text is rectified using the orientation of text-line and vertical strokes. The proposed method outperforms state-of-the-art algorithms in English as well as Asian characters in the extensive experiments.

Key words: document image, document rectification, document dewarping, document stitching, generalized cylindrical surface, text-line, scene text detection

Student number: 2008-20836

Contents

Abstract **i**

Contents **iv**

List of Figures **vii**

List of Tables **xii**

1 Introduction **1**

1.1 Document rectification via text-line based optimization 2

1.2 A unified approach of rectification and stitching for document images 4

1.3 Rectification via scene text detection 5

1.4 Contents 7

2 Related work **9**

2.1 Document rectification 9

2.1.1 Document dewarping without text-lines 9

2.1.2 Document dewarping with text-lines 10

2.1.3 Text-block identification and text-line extraction 11

2.2 Document stitching 12

2.3 Scene text detection 13

3 Document rectification based on text-lines 15

3.1 Proposed approach 15

3.1.1 Image acquisition model 16

3.1.2 Proposed approach to document dewarping 18

3.2 Proposed cost function and its optimization 22

3.2.1 Design of $E_{str}(\cdot)$ 22

3.2.2 Minimization of $E_{str}(\cdot)$ 23

3.2.3 Alignment type classification 28

3.2.4 Design of $E_{align}(\cdot)$ 29

3.2.5 Design of $E_{spacing}(\cdot)$ 31

3.3 Extension to unfolded book surfaces 32

3.4 Experimental result 34

3.4.1 Experiments on synthetic data 36

3.4.2 Experiments on real images 39

3.4.3 Comparison with existing methods 43

3.4.4 Limitations 45

4 Document rectification based on feature detection 49

4.1 Proposed approach 50

4.2 Proposed cost function and its optimization 51

4.2.1 Notations 51

4.2.2 Homography between the i-th image and $\boldsymbol{\pi}_E$ 52

4.2.3 Proposed cost function 53

4.2.4 Optimization 53

4.2.5 Relation to the model in [17] 55
4.3 Post-processing 55
4.3.1 Classification of two cases 56
4.3.2 Skew removal 56
4.4 Experimental results 57
4.4.1 Quantitative evaluation on metric reconstruction performance 57
4.4.2 Experiments on real images 58

5 Scene text detection and rectification **67**
5.1 Introduction 67
5.1.1 Contribution 67
5.1.2 Proposed approach 69
5.2 Candidate region detection 70
5.2.1 CC extraction 70
5.2.2 Computation of similarity between CCs 70
5.2.3 CC clustering 73
5.3 Rectification of candidate region 73
5.4 Text/non-text classification 76
5.5 Experimental result 80
5.5.1 Experimental results on ICDAR 2011 dataset 80
5.5.2 Experimental results on the Asian character dataset 80

6 Conclusion **83**

Bibliography **87**

Abstract (Korean) **97**

List of Figures

1.1 Examples of Korean Text. In case of Korean, it is hard to divide CCs into each text line without language model 6

2.1 Text-line representation. (a) Continuous text-line models (polynomials or B-splines) used by conventional methods and (b) text-line and text-block representation [3] adopted in this paper. 11

3.1 Image acquisition model. (a) document in an imaginary uv-plane, (b) GCS surface, (c) camera and document surface, (d) camera-captured image. 16

3.2 The first column: input images, the second column: results using only straight line constraints, the third column: results using straight line constraints and equal line-spacing constraints, and the fourth column: results using straight line constraints and aligned text-block constraint. 19

3.3 Decision boundaries for alignment type classification. 28

3.4 Projective transformation between two planar surfaces. 31

3.5 Examples of synthetic images. 35

3.6 Rectification results of planar documents. Images in the first column are inputs, where colored circles are CCs in detected text-lines and red lines show the sides of each text-block. Images in the second column are the rectified images and regions inside red rectangles are enlarged in the third column. Figures in the last column show reconstructed document surfaces and cameras. 40

3.7 Rectification results of curved documents. Images in the first column are inputs, where colored circles are CCs in detected text-lines and red lines show the sides of each text-block. Images in the second column are the rectified images and regions inside red rectangles are enlarged in the third column. Figures in the last column show reconstructed document surfaces and cameras. 41

3.8 Rectification results of book images. Images in the first column are inputs, and images in the second column are text-line extraction results where colored circles are CCs in detected text-lines and red lines show the sides of each text-block. Images in the third column are the rectified images where the red lines indicate book bindings. Figures in the last column show reconstructed document surfaces and cameras. 42

3.9 Dewarping results on CBDAR2007 data set. From left to right column, input image, scanned ground truth and the results obtained by SEG, SKEL, CTM, snakes, and the proposed method are shown. . . 44

3.10 Box plot of the rectification error of six methods. The rectification errors of six methods are measured on CBDAR2007 data set (some results are from [4]). 46

3.11 Box plot of the OCR accuracy of six methods. The OCR accuracies of six methods are measured on CBDAR2007 data set (some results are from [4]). 47

4.1 Results of the proposed algorithm. (a), (b) Input images, (c), (d) Estimated camera poses, (e), (f) Final results. 60

4.2 Flowchart of the proposed algorithm. 61

4.3 Comparison between the sequential registrations and the proposed method. (a) Eight input images capturing the same plane, (b) Document stitching result using the pairwise homographies. Note that the composite suffers from error accumulations, (c) Document stitching result using the proposed method. The proposed method minimizes the global registration errors and it less suffers from mis-registrations. 62

4.4 Illustration of notations in this paper. 63

4.5 Up-vector computation for skew removal in composites. (a) Two input images, (b) Visualization of estimated parameters, (c) Illustration of a up-vector **u**. 63

4.6 (a), (b) Synthesized image pair, (c) Average value of (4.32) for the 275 pairs of synthesized images according to the margin of error. . . 64

4.7 Comparison of the proposed result with the conventional image stitching method. (a) Six input images, (b) Result of *Autostitch*, (c) Result of the proposed method. 64

4.8 Image stitching result on the images captured by a moving camera. (a) Nine input images, (b) Estimated camera poses, (c) Image stitching result that considers the fourth image as a reference, (d) Image stitching result using the proposed method. 65
4.9 Image stitching results for document images. (a), (c) Input images, (b), (d) Image stitching results of the proposed method. 66

5.1 Examples of Asian Text. In case of Asian character, it is hard to divide CCs into each text line without language model 68
5.2 (The flowchart of the proposed method.) 69
5.3 An example of scene text image. (a) original image (b) binarization result (c) MSER result 71
5.4 Illustration of geometrical relations between CCs 72
5.5 Result of the proposed clustering method. (a) Original image (b) MSER result (c) Coarse clustering result (d) Refined clustering result. 74
5.6 Rectification of the detected text box. (a) Merged bounding boxes of CCs. Red circles is the center points of top lines, and green circles is the center points of bottom lines (b) Estimated boundaries of text box (c) Rectified images using the estimated boundaries. 77
5.7 Examples of mis-classification with a single MLP classifier. The detected text boxes are displayed as red rectangles. 78
5.8 Proposed two MLP classifiers. 79
5.9 Results of the proposed method in the Asian character data set. Red rectangles are the detected text box. 82

List of Tables

3.1 Mean rectification error with respect to the norm of rotation vector. Errors are measured in the unit of pixel and the values in the parentheses are the standard deviations. 36

3.2 Mean rectification error with respect to surface complexities. Errors are measured in the unit of pixel and the values in the parentheses are the standard deviations. 37

3.3 Mean rectification error with respect to noise in text-lines. Errors are measured in the unit of pixel and the values in the parentheses are the standard deviations. 38

5.1 Evaluation on ICDAR 2011 dataset 80

5.2 Evaluation on Asian character dataset 81

Chapter 1

Introduction

One of the most important information in our daily lives is the text information such as road signs, books, documents, labels, etc. Humans can easily locate the text and recognize the meaning of it without much efforts. This is because humans have naturally learned what text does look like and how text on the documents or in the scene is deformed depending on viewing direction, lighting, or shape of background from a early age. However, enabling a machine to detect and recognize text is not an easy problem. One of representative researches in this area is the optical recognition of printed material, which converts printed documents into digital images by using flatbed scanners, and then performs layout analysis and optical character recognition (OCR) [1, 2]. However, as mobile (or wearable) devices equipped with high-performance digital cameras are becoming widely available, many researchers tried to replace flatbed scanners with digital cameras [3, 4]. Digital cameras have a number of advantages over flatbed scanners, such as portability, fast response, non-contact property (no physical damages on documents), and unconstrained view point. However, camera-captured images usually suffer from geometric distortions:

(i) perspective distortions in the camera-capturing process and (ii) non-linear distortions caused by the curved surfaces of documents. These make the camera-captured document images deviate from the flat surface so that it becomes difficult to recognize the document layout, text, and any other information from them. The general approach to understanding document from the camera-captured image is to remove such distortions using several easy-to-extract information before recognition. This process of removing the distortions from the document image is called the document rectification. In this dissertation, two new document rectification algorithms via optimizations of cost functions based on easy-to-extract information are proposed and an efficient and robust scene text detection algorithm is also provided. The extracted scene text can be used for rectification and recognition.

1.1 Document rectification via text-line based optimization

The document rectification or dewarping is a process to recover 2D texture of original documents from the camera-captured documents. Documents are usually printed on thin papers which are vulnerable to bending, therefore capturing original contents of the printed documents without any distortion is not an easy process. Besides the distortion caused by bending, even a slight misalignment between the camera and the document surface induces perspective distortion. Hence camera-captured documents images show lower visual qualities than those of flatbed scanners. To improve the qualities of them to the level of images from the flatbed scanners, a number of approaches have been proposed [5–14, 4]. Among various approaches, text-line based approaches have received a lot of attentions, because text-lines are

the most salient properties of documents and this approach allows us to develop an algorithm that needs a single image as the input.

In this dissertation, a new document dewarping algorithm that also exploits the text-lines is proposed. Although there are many methods based on the text-lines as mentioned above, the proposed method is different from the conventional ones in several aspects. First, the dewarping problem is formulated as an optimization problem based on the given text-line and text-block information. Note that many algorithms consist of a series of steps such as vanishing point estimation, aspect ratio estimation, etc, where each step assumes that the previous steps are successful. On the other hand, the same problem is addressed by optimizing a single cost function, whose minimization yields a set of parameters enabling the reduction of geometric distortions. Second, the proposed formulation does not require continuous text-line models (i.e., it does not need to fit the baselines with polynomials or splines), so that an existing text-block identification and text-line extraction method in [3] can be used for the feature extraction. Third, the proposed method does not need special assumption on documents layouts (such as single column format or justified page layout). Rather, the proposed method detects text-blocks and imposes certain constraints according to their properties, which gives the robustness to the variation of layouts and background. Finally, the proposed method deals with not only planar documents but also curved surfaces in the same framework, which has been treated in a different manner [4, 14]. Moreover, the proposed method can be extended to the unfolded book surfaces by changing the surface models.

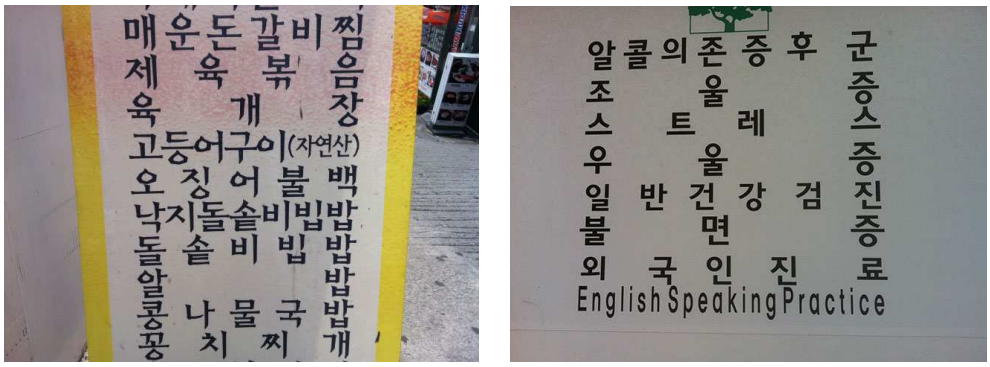
1.2 A unified approach of rectification and stitching for document images

The second one is a unified method of rectification and stitching for planar documents using multiple images. The image stitching is a process that generates a high-resolution and/or large field-of-view (FoV) image from the overlapping views. In general, the stitching algorithms are based on a (planar) homography model, which is a geometric relationship between two images when (i) the optical center of a camera is fixed or (ii) the images contain the same plane [15]. Between the above two conditions allowing image stitching, most image stitching methods focused on the first one (fixed-optical-center case) [15,16]; it is probably because this condition is (easily) satisfied when users take pictures of distant targets and this also simplifies camera models. However, this condition is not taken for granted in capturing a document since the document is fixed on desk or table and the camera is moving. To deal with this kind of camera motion, a new parametric family of motion model is proposed in this dissertation. It is derived not from the simplified camera model but from the full camera model, which involves the locations and poses of all the cameras. For estimating the camera position and viewing angle, general feature points are employed here instead of text-lines utilized in the first method. If several images captured from different view points are available, correspondence between feature points can be computed. These correspondences impose the constraints on the camera position and camera pose. By solving the constraints, the camera pose can be estimated and finally the document image can be rectified. Besides fixed camera condition, the general image stitching algorithms [16,17] have no constraints on the reference plane (where the document is laid on), so that they cannot locate

physically correct reference plane in the world coordinates. To address the problem, a new cost function is proposed to impose the constraints on the reference plane. By optimizing the cost function, the correct reference plane can be estimated without prior knowledge. The estimated reference plane can also be used to rectify the composite image. Furthermore, the proposed method can be applied to any other planar object such as building facades or mural paintings as well as the camera-captured document image since it employs the general features.

1.3 Rectification via scene text detection

In recent years, high-quality cameras have been employed in the mobile devices, and this brings a variety of camera applications such as face detection, image stitching, or augmented reality(AR) to users. Among them, detection or recognition of text in the natural scene is attaining lots of attention because text has much information and can be used in various application. For example, the detected text can be used to tag the picture, can be helpful for the visually impaired, or can be translated to other languages. In addition, the detected text can be clustered into text-lines, and they can be used to rectify the document or the plane. Recently scene text detection methods using connected component (CC) are proposed in [18,19]. These approaches assume that a character consists of a single CC like English alphabet, however, this assumption is brittle in the Asian characters such as Korean, Chinese, and Japanese. Especially, a single Korean character is comprised of three CCs, i.e., initial consonant, vowel, and a consonant placed under a vowel. Moreover, some last consonants may be located closer to the first consonant in the lower text line than the their initial consonant or vowel as shown in Fig 1.1, therefore it is difficult to divide



(a) (b)

Figure 1.1: Examples of Korean Text. In case of Korean, it is hard to divide CCs into each text line without language model

CCs into each text line without language model in the Asian characters. Usually, existing methods assume only the relation between characters without considering inter-character relations, and thus they fail to cluster CCs into text lines in the Asian characters [18, 20, 19].

To alleviate the problem, a new clustering method is proposed based on the similarity measure considering inter-character relation. The proposed method consists of candidate generation, spectral clustering of each region, rectification of clusters, and text/non-text classification. For candidate generation step, the maximally stable extremal region (MSER) algorithm is employed [21]. Motivated from [19], the proposed method clusters the candidate regions by measuring the adjacency between text regions, and the verification of each cluster based on the simple rules is followed. The adjacency measure is trained on the dataset labeled with the bounding box of text region. This "text/non-text classification first" is more efficient than complex discriminative classifier adopted in the previous method [22–24]. The detected text regions in the clustering step still contain an amount of non-text regions,

hence the second text/non-text classifier filters out the remaining non-text region. To improve the accuracy of the classifier, the detected text regions are rectified using the orientation of text-line and the vertical strokes before classification. In this step, optimization-based vertical stroke estimation method is employed. The second classifier adopts more complex algorithm i.e., multi-layer perceptron (MLP) than the previous classifier. In addition to the MLP classifier used in [19] which takes only a single sub-region into account, the proposed method employs the additional MLP classifier considering pair-wise relations between sub-regions of the detected text. Combined with the proposed rectification step, this improves the precision of the detected regions. The proposed method achieves the state-of-the-art results in Korean as well as English in the standard dataset.

1.4 Contents

In chapter 2, related works are introduced. Firstly, the conventional approaches to document rectification are reviewed. In general, there are two approaches in the document rectification depending on the utilization of text-lines. The rectification methods without text-lines are reviewed, and the approaches using text-lines are followed. The text-line extraction methods and their related issues are also provided. In chapter 3, the proposed document rectification method based on text-lines are introduced. In this chapter, the image acquisition model are introduced and the rectification procedure following the image acquisition process in a reverse manner are described. After that, the proposed cost function using the text-lines and the text-blocks are introduced. The extension to unfolded book surface are described and experimental results are followed. In chapter 4, the unified approach of the

rectification and the stitching for the planar surfaces is presented. The proposed cost function is described and experimental results are presented. In chapter 5, the proposed scene text detection method with the rectification of text are described. Firstly, the similarity function used in the spectral clustering is introduced, and training method to build the similarity function is explained. Text/non-text classifier combined with the proposed rectification step is described and the experimental results are followed. Finally, this dissertation is concluded in chapter 6.

Chapter 2

Related work

2.1 Document rectification

Although there were some model-free approaches to rectifying images by straightening individual words and text-lines [5,6], most of work assumed geometric models of document surfaces and tried to find global transformations that rectify camera-captured images. In this section, we first review various rectification methods and discuss issues in text-line based approaches.

2.1.1 Document dewarping without text-lines

Since the 3D structure of an object can be estimated by structured light or laser scanner, many document rectification methods were developed based on this kind of shape measuring hardware [7, 8]. Although these approaches allow us to handle a range of surface distortions and yield highly accurate results, requirements on special equipments are their major drawbacks. We can also reconstruct 3D structures from multiple images, and some methods are based on these shape-from-motion

approaches [25, 26]. Although they can rectify pages without specialized hardwares, their high complexities and requirements on multiples images are major problems. The 3D shape information of documents can also be estimated from a single image with special assumptions. For example, under controlled lighting conditions, we can reconstruct document surfaces with shape-from-shading approaches [9, 10]. While they provide theoretically sound rectification frameworks, their strict assumptions on illuminations are not practical in many situations. There is also another kind of single-view based method that rectifies images by exploiting the regularity of textures [27]. However, there are limitations in dealing with documents having complex layout and non-text regions, where textures are no longer regular.

2.1.2 Document dewarping with text-lines

Among various approaches to document image dewarping, the most popular one may be the text-line based method. It is because text-lines are the most common properties of documents and this approach enables a single-view algorithm. Among many text-line based methods, the authors in [11, 12] considered text-lines as top and bottom boundaries and tried boundary interpolations, which has been used for the rectification of pictures under the generalized cylindrical surface (GCS) assumptions [25, 28]. This approach is efficient, however, it lacks theoretical justification and it requires assumptions such as single-column and justified pages. Some methods estimated vertical directions as well as text-lines [13, 14]. Since 2D text-field is available in these approaches, they less suffer from ambiguities and are able to handle complex document surfaces [13]. However, building 2D text-field from document images (possibly having complex layouts, non-textual objects) is a more challenging problem than text-line extraction. Recently, the authors in [4] showed that GCS

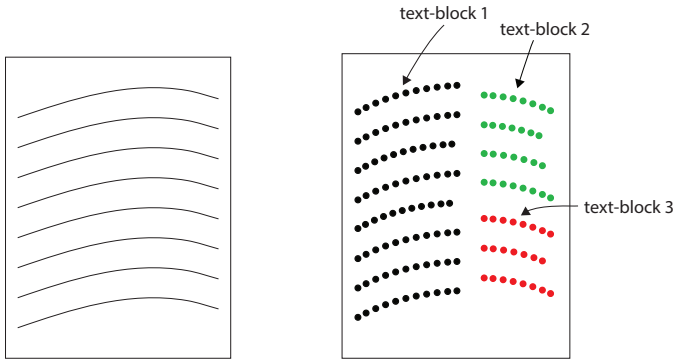


Figure 2.1: Text-line representation. (a) Continuous text-line models (polynomials or B-splines) used by conventional methods and (b) text-line and text-block representation [3] adopted in this paper.

surfaces can be reconstructed from horizontal text-lines only (with mild approximations). Even though its theoretical backgrounds are sound, its sensitivity to the accuracy of text-lines is not well addressed. Also, this approach is valid only for curved single-column pages.

2.1.3 Text-block identification and text-line extraction

Text-block identification (text-region segmentation) and text-line extraction are essential for many document dewarping algorithms. However, many conventional methods did not address these issues, rather they assumed user interactions for border noise removal or started from extracted text regions [4, 14]. Also, conventional dewarping methods require continuous text-line models as illustrated in Fig. 2.1-(a), and they find text-lines by fitting the fiducial points of characters with polynomials [4]. The accuracy of this fitting procedure is one of the key factors for the success of existing text-line based methods [11, 13, 4], however, the sensitivity with

respect to the fitting accuracy is not clearly understood.

2.2 Document stitching

Before reviewing the document stitching, it needs to look over the work related with general image stitching since the document stitching is usually studied as a specific application of the general image stitching algorithms. While there are many works in the image stitching area, the representative work is a fully automatic system proposed in [17], which discovers matching relationship between the images and recognizes panoramas automatically. In [17], each camera (corresponding to each image) is parameterized with four variables (three for the rotation matrix and one for the focal length) and they were estimated by minimizing pairwise registration errors. The method was shown to be very robust to photometric variations and non-ideal effects (e.g., radial distortions, moving objects, slight violations of the assumption, an so on).

Unlike the image stitching, conventional document stitching algorithms consist of two step, the first step is to remove the perspective distortion and the second step is to find corresponding points and stitch the images [29–32]. To rectify camera-captured document images, the previous methods use variance of the pixel intensity [29], text line and vertical character stroke [30, 31], vanishing points [32]. However, these approaches has the limitation in dealing with non-text document such as wall painting and large document which is hard to estimate vanishing points using boundary of documents. The algorithm to mosaic the two images based on comparing values of pixels in the overlapping split images is also developed [33]. However, this method needs images without perspective distortion and is not appropriate for

mosaicing the more than two images. The algorithms to perform rectification and stitching by estimating camera motion exist [34, 35]. Although they can estimate camera pose and shape of document surfaces without special assumptions, their requirements of dense overlapped video and high computational cost required to process a number of frames are major problems.

2.3 Scene text detection

In general, scene text detection algorithm can be classified to two main approaches, sliding-window based and connected component (CC) based approaches [22–24]. The sliding-window based algorithm [36,37] divide the whole image into sub-images and determine whether each sub-image contain text or text-like object, which is motivated from other object detection algorithm such as face detection [38]. This approach is robust to noise and blur, but is not efficient when considering the scale and rotation because the number of sub-image that needs to evaluate increase massively. Moreover, designing an efficient and robust classifier is also challenging since even for human it is difficult to determine whether small patch is a part of text or not without prior information about language, size, rotation, and text color. In conventional approaches, cascade structures based on simple features like horizontal and vertical derivatives are employed for efficiency [36, 36, 39]. The CC based approaches localize text region by connecting the region with the similar property and position. The used properties are stroke width [40, 18], intensity, color [21, 19], or edge response [20]. Recently the most researches in this area adopt the CC based approach. This method has an advantage that the complexity is independent with the scale and rotation of text. Moreover, the efficient CC clustering algorithm also

exist [21]. The main drawback of this method is the assumption that a character is a single CC. The detected CCs are usually noisy so that they may be a part of certain character or a merging of several characters. To prevent the noisy CCs, a variant of maximally stable extremal region (MSER) algorithm is proposed in [41, 42].

Chapter 3

Document rectification based on text-lines

3.1 Proposed approach

Text-block identification and text-line extraction in camera-captured document images are challenging problems due to perspective distortions, geometric distortions, and non-textual objects (including background clutters). In order to deal with these challenges, a robust method in [3] is adopted. This algorithm is able to handle challenging cases by exploiting periodic properties of text-blocks, and provides information on text-blocks as well as text-lines as shown in Fig. 2.1-(b) (see also the first columns of Figs. 3.6–3.7 and the second column of Fig. 3.8). Unlike conventional methods that tried to fit text-lines as continuous models [11, 13, 4], the proposed method works on connected components (CCs) directly and represents results provided by [3] as

$$\mathcal{P} = \{p_i^{k,j} \in \Re^2\} \tag{3.1}$$

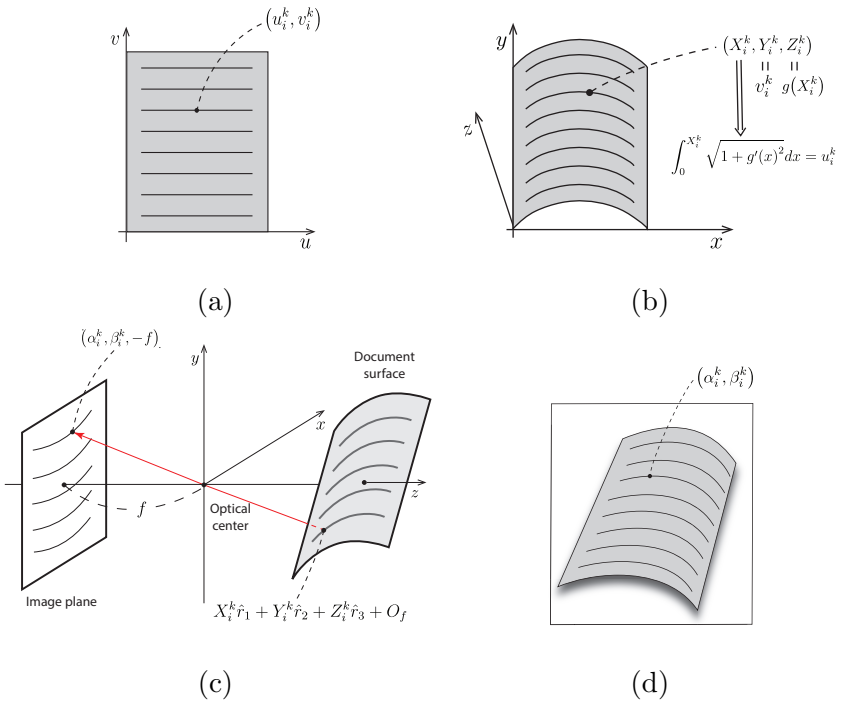


Figure 3.1: Image acquisition model. (a) document in an imaginary uv-plane, (b) GCS surface, (c) camera and document surface, (d) camera-captured image.

where $p_i^{k,j}$ is the center of the i-th CC in the k-th text-line in the j-th text-block. Although this method deals with multiple blocks, for the simplicity of presentation, the document is assumed to have only a single block: p_i^k is used rather than $p_i^{k,j}$ by omitting the block index j. Since the proposed cost function is additive, the extension to multiple text-blocks is straightforward.

3.1.1 Image acquisition model

The document surfaces are modeled as GCSs like many conventional work [43, 28, 44, 4]. That is, a point (u, v) in an imaginary uv-plane (Fig. 3.1-(a)) is transformed

to

$$S(u, v) = [x \ y \ g(x)]^T \tag{3.2}$$

where

$$u = \int_0^x \sqrt{1 + g'(t)^2} dt. \tag{3.3}$$

$$v = y \tag{3.4}$$

as shown in Fig. 3.1-(b). Here, it is assumed that document surfaces can be expressed with polynomials, i.e.,

$$g(x) = \sum_{m=0}^{M} a_m x^m. \tag{3.5}$$

Note that this model is able to deal with planar surfaces as well as curved ones, by putting $g(x) \equiv 0$. This surface is captured by a camera as shown in Fig. 3.1-(c) and the document image is obtained as in Fig. 3.1-(d). As shown in Fig. 3.1-(c), it is assumed that the origin of the camera coordinate system is located at $(0, 0, 0)$ in the world coordinate system, the camera sensor is perpendicular to z-axis, and it passes $(0, 0, -f)$. The pose of a document surface is given by similarity transforms

$$X(u, v) = s\mathbf{R}^T S(u, v) + t \tag{3.6}$$

where $s \in \Re$ is a scaling factor, $\mathbf{R}$ is a rotation matrix, and $t \in \Re^3$ is a translation vector. Finally, a standard pin-hole camera model [15] is adopted, where a principal point is at the center of the image and its focal length is f. In summary, a point (u, v) in the imaginary plane corresponds to the point in the image plane

$$(\alpha, \beta) = \left(-f \frac{X(u, v) \cdot \hat{e}_1}{X(u, v) \cdot \hat{e}_3}, -f \frac{X(u, v) \cdot \hat{e}_2}{X(u, v) \cdot \hat{e}_3} \right) \tag{3.7}$$

where $\hat{e}_i$ are unit vectors representing three axes ($i = 1, 2, 3$) in the world coordinate system.

Many parameters are involved in the overall process, however, it is not necessary to estimate all of them, since different parameters may result in equivalent results (up to scale and translation). Algebraic manipulations show that the changes of s and t yield equivalent images (the surface function $g(x)$ is also changing). Therefore, without loss of generality, s and t may be set as $s = 1$ and $t = O_f = [0\ 0\ f]^T$. Note that this choice yields rectified images having similar resolution to inputs. Based on this setting, the document surface is formulated as

$$X(u, v) = x\hat{r}_1 + y\hat{r}_2 + g(x)\hat{r}_3 + O_f \tag{3.8}$$

where $\hat{r}_m$ is the m-th column vector of the $\mathbf{R}^T$. For the parametrization of $\mathbf{R}$, the exponential representations is employed: for $\Theta = [\theta_1, \theta_2, \theta_3]^T$

$$\mathbf{R}(\Theta) = \exp\left([\Theta]_\times\right) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{3.9}$$

where

$$[\Theta]_\times = \begin{bmatrix} 0 & -\theta_3 & \theta_2 \\ \theta_3 & 0 & -\theta_1 \\ -\theta_2 & \theta_1 & 0 \end{bmatrix}. \tag{3.10}$$

3.1.2 Proposed approach to document dewarping

In the proposed model, the document dewarping can be considered a process to estimate Θ and $\{a_m\}$ from $\mathcal{P}$. This process is formulated as an optimization problem and a cost function is designed by encoding the properties of rectified documents. Text-lines in the rectified images should be straight in the horizontal direction. In other words, when two points are on the same text-line, their corresponding points

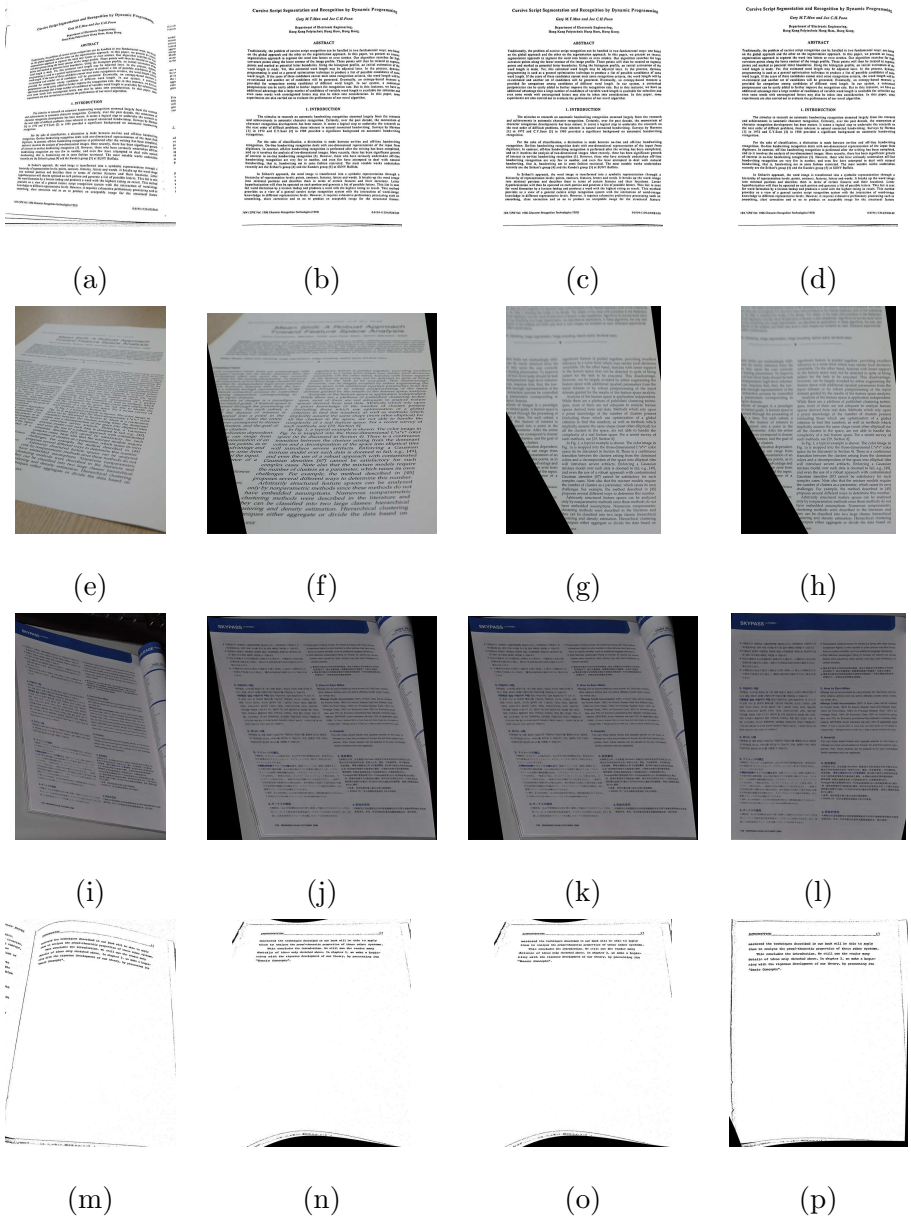


Figure 3.2: The first column: input images, the second column: results using only straight line constraints, the third column: results using straight line constraints and equal line-spacing constraints, and the fourth column: results using straight line constraints and aligned text-block constraint.

in the uv-plane have the same v-value. Based on this observation, a cost function is developed as (details of this function will be presented in the next section)

$$E_0(\cdot) = E_{str}(\cdot) \tag{3.11}$$

where the subscript *str* addresses that this cost is related with the straightness of result. Document dewarping results using this function are shown in the second column in Fig. 3.2. As shown in Fig. 3.2-(b), this constraint allows us to rectify images when surfaces are curved and there are a number of text-lines. However, this constraint is not enough in some cases. For planar surfaces as shown in Fig. 3.2-(f) and (j), the results still suffer from perspective distortions: text-lines in both of Fig. 3.2-(f) and (g) are horizontally straight and it is hard to determine whether (g) is a better solution than (f) with this cost function.

In order to resolve this ambiguity, properties of text-blocks are exploited. In the same text-block, line-spacing between two neighboring text-lines should be constant, which can be encoded into the cost function by adding a new term:

$$E_1(\cdot) = E_{str}(\cdot) + E_{spacing}(\cdot) \tag{3.12}$$

where $E_{spacing}(\cdot)$ imposes constraints on line-spacings. Experimental results using this function are shown in the third column in Fig. 3.2. As shown in Fig. 3.2-(g), this constraint resolves the ambiguities and yields improved solutions, however, there are still limitations. In the case of the third row in Fig. 3.2, the image has a large focal length (i.e., $f \gg 1$) and its transform is well approximated with an affine transform (weak-perspective situations). In affine transforms, the ratio of lengths on collinear lines is invariant, and $E_{spacing}(\cdot)$ does not have discriminating powers to determine whether (l) is a better solution than (k). Also, this condition is too

subtle to deal with a text-block having a small number of text-lines as shown in Fig. 3.2-(o).

Besides equal line-spacing, most text-block is either left-aligned, right-aligned, or justified since people feel visual comforts on the well-aligned text-blocks. These are very strong constraints and problems of (3.12) can be resolved based on them. The cost function using the aligned text-block is denoted as

$$E_2(\cdot) = E_{str}(\cdot) + E_{align}(\cdot) \tag{3.13}$$

where $E_{align}(\cdot)$ encodes constrains on the text-block alignment. Results using this cost function are shown in the last column in Fig. 3.2, which is shown to alleviate the above stated problems.

Although $E_2(\cdot)$ is a very effective cost function, it cannot be used without the knowledge on alignment types. Hence, an alignment type classification method for text-blocks is also developed. At the first glance, this classification seems like a very difficult problem, because it suffers from the same challenges discussed in text-line extractions. However, this problem can be dealt with efficiently and effectively, using the information from the optimization of $E_{str}(\cdot)$. Specifically, the minimization of $E_{str}(\cdot)$ yields the rough estimates of distortion parameters, and they enable the reduction of geometric distortions caused by page curls and perspective distortions (except the effect of pitch angles) as shown in the second column of Fig. 3.2. Therefore, the type of blocks is determined with simple rules (the details of the classification method will be presented in the next section). Moreover, the classification accuracy need not be very high. As shown in Fig. 3.2, the cost function in (3.12) works without type information and yields reasonable results. Therefore, the cost function in (3.12) is basically used, or it is switched to (3.13) when block types are

clearly determined.

Fortunately, the proposed switching idea based on the rough rectification result with (3.11) does not introduce noticeable overhead compared with approaches minimizing either (3.12) or (3.13). These two cost functions are highly non-linear, and its direct minimization is likely to be stuck to local minima. In order to avoid local minima, a coarse solution has to be found using (3.11) and the result is refined by minimizing either (3.12) or (3.13).

3.2 Proposed cost function and its optimization

In this section, the details of the proposed cost function are presented and its minimization method is discussed. Since determining the type of alignments is required for the cost function switching, a classification algorithm is also presented.

3.2.1 Design of $E_{str}(\cdot)$

Let us assume that $p_i^k = (\alpha_i^k, \beta_i^k) \in \mathcal{P}$ is the i-th point on the k-th text-line in a captured image as shown Fig. 3.1-(d). Then, its corresponding point on the document surface can be obtained by computing the intersection between a ray from $\tilde{p}_i^k = \left(\alpha_i^k, \beta_i^k, -f\right)$ and the document surface (see Fig. 3.1-(c)). This ray equation is

$$P(t) = \tilde{p}_i^k \times t \tag{3.14}$$

for $t \in \Re$, and the intersection between the document surface (3.8) and the ray is given by

$$P(t_i^k) = X_i^k \hat{r}_1 + Y_i^k \hat{r}_2 + Z_i^k \hat{r}_3 + O_f \tag{3.15}$$

where

$$g\Big(X_i^k\Big) = Z_i^k. \tag{3.16}$$

To be precise,

$$\begin{bmatrix} X_i^k \\ Y_i^k \\ Z_i^k \end{bmatrix} = R \times \left(\tilde{p}_i^k t_i^k - O_f \right) = \begin{bmatrix} \hat{r}_1^T \\ \hat{r}_2^T \\ \hat{r}_3^T \end{bmatrix} \left(\tilde{p}_i^k t_i^k - O_f \right). \tag{3.17}$$

In order to find the intersection point, it is required to solve (3.16) and (3.17), which is a polynomial equation of degree M and can be solved by using the Newton-Raphson method [45]. Since Y_i^k is the v-value in the rectified plane, the first cost function is given by

$$E_{str}\Big(\Theta, \{a_m\}, \{l^k\}\Big) = \sum_k \sum_i (Y_i^k - l^k)^2 \tag{3.18}$$

where l^k is a new parameter representing the vertical position of the k-th text-line.

3.2.2 Minimization of $E_{str}(\cdot)$

The cost function (3.18) is given by the sum of squares and it can be optimized with the Levenberg-Marquardt algorithm [46]. For performing the Levenberg-Marquardt algorithm, the derivatives of

$$E(k,i) = Y_i^k - l^k \tag{3.19}$$

with respect to each parameter are derived as:

$$\frac{\partial E(k,i)}{\partial l_m} = -\delta(k,m) \tag{3.20}$$

$$\frac{\partial E(k,i)}{\partial \theta_1} = \frac{\partial Y_i^k}{\partial \theta_1} = \left(\frac{\partial \hat{r}_2}{\partial \theta_1}\right)^T \tilde{p}_i^k t_i^k + \hat{r}_2^T \tilde{p}_i^k \frac{\partial t_i^k}{\partial \theta_1} - \frac{\partial r_{23}}{\partial \theta_1} f \tag{3.21}$$

$$\frac{\partial E(k,i)}{\partial \theta_2} = \frac{\partial Y_i^k}{\partial \theta_2} = \left(\frac{\partial \hat{r}_2}{\partial \theta_2}\right)^T \tilde{p}_i^k t_i^k + \hat{r}_2^T \tilde{p}_i^k \frac{\partial t_i^k}{\partial \theta_2} - \frac{\partial r_{23}}{\partial \theta_2} f \tag{3.22}$$

$$\frac{\partial E(k,i)}{\partial \theta_3} = \frac{\partial Y_i^k}{\partial \theta_3} = \left(\frac{\partial \hat{r}_2}{\partial \theta_3}\right)^T \tilde{p}_i^k t_i^k + \hat{r}_2^T \tilde{p}_i^k \frac{\partial t_i^k}{\partial \theta_3} - \frac{\partial r_{23}}{\partial \theta_3} f \tag{3.23}$$

$$\frac{\partial E(k,i)}{\partial a_m} = \hat{r}_2^T \tilde{p}_i^k \frac{\partial t_i^k}{\partial a_m}. \tag{3.24}$$

$$\tag{3.25}$$

Detailed derivations of $\frac{\partial \hat{r}_2}{\partial \theta_l}$, $\frac{\partial t_i^k}{\partial \theta_l}$ $(l = 1, 2, 3)$ and $\frac{\partial t_i^k}{\partial a_m}$ are described in the following sections.

Derivative of rotation matrix

From Rodrigues' rotation formula, the rotation matrix is computed by

$$\mathbf{R}(\Theta) = \exp\left([\Theta]_\times\right) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} 1 - \left(\theta_2^2 + \theta_3^2\right) s & -\theta_3 r + \theta_1\theta_2 s & \theta_2 r + \theta_1\theta_3 s \\ \theta_3 r + \theta_1\theta_2 s & 1 - \left(\theta_1^2 + \theta_3^2\right) s & -\theta_1 r + \theta_2\theta_3 s \\ -\theta_2 r + \theta_1\theta_3 s & \theta_1 r + \theta_2\theta_3 s & 1 - \left(\theta_1^2 + \theta_2^2\right) s \end{bmatrix} \tag{3.26}$$

where $r = \sin\left(\|\Theta\|\right) / \|\Theta\|$ and $s = \left(1 - \cos\left(\|\Theta\|\right)\right) / \|\Theta\|^2$.

Derivatives of R with respect to θ_1 is given by

$$\frac{\partial r_{11}}{\partial \theta_1} = -\left(\theta_2^2 + \theta_3^2\right)\frac{\partial s}{\partial \theta_1} \tag{3.27}$$

$$\frac{\partial r_{12}}{\partial \theta_1} = -\theta_3\frac{\partial r}{\partial \theta_1} + \theta_2 s + \theta_1\theta_2\frac{\partial s}{\partial \theta_1} \tag{3.28}$$

$$\frac{\partial r_{13}}{\partial \theta_1} = \theta_2\frac{\partial r}{\partial \theta_1} + \theta_3 s + \theta_1\theta_3\frac{\partial s}{\partial \theta_1} \tag{3.29}$$

$$\frac{\partial r_{21}}{\partial \theta_1} = \theta_3\frac{\partial r}{\partial \theta_1} + \theta_2 s + \theta_1\theta_2\frac{\partial s}{\partial \theta_1} \tag{3.30}$$

$$\frac{\partial r_{22}}{\partial \theta_1} = -2\theta_1 s - \left(\theta_1^2 + \theta_3^2\right)\frac{\partial s}{\partial \theta_1} \tag{3.31}$$

$$\frac{\partial r_{23}}{\partial \theta_1} = -r - \theta_1\frac{\partial r}{\partial \theta_1} + \theta_2\theta_3\frac{\partial s}{\partial \theta_1} \tag{3.32}$$

$$\frac{\partial r_{31}}{\partial \theta_1} = -\theta_2\frac{\partial r}{\partial \theta_1} + \theta_3 s + \theta_1\theta_3\frac{\partial s}{\partial \theta_1} \tag{3.33}$$

$$\frac{\partial r_{32}}{\partial \theta_1} = r + \theta_1\frac{\partial r}{\partial \theta_1} + \theta_2\theta_3\frac{\partial s}{\partial \theta_1} \tag{3.34}$$

$$\frac{\partial r_{33}}{\partial \theta_1} = -2\theta_1 s - \left(\theta_1^2 + \theta_2^2\right)\frac{\partial s}{\partial \theta_1} \tag{3.35}$$

$$\tag{3.36}$$

where

$$\frac{\partial r}{\partial \theta_1} = \frac{\cos\left(\|\Theta\|\right)\theta_1 - \sin\left(\|\Theta\|\right)\frac{\theta_1}{\|\Theta\|}}{\|\Theta\|^2} \tag{3.37}$$

$$\frac{\partial s}{\partial \theta_1} = \frac{\sin\left(\|\Theta\|\right)\theta_1\|\Theta\| - \left(1-\cos\left(\|\Theta\|\right)\right)2\theta_1}{\|\Theta\|^4}. \tag{3.38}$$

$\frac{\partial \mathbf{R}}{\partial \theta_2}$ and $\frac{\partial \mathbf{R}}{\partial \theta_3}$ can be obtained in the same manner.

Derivatives of t_i^k with respect to rotation angles

In order to compute $\frac{\partial t_i^k}{\partial \theta_m}$, both sides of (3.16) need to be differentiated:

$$g'\left(X_i^k\right)\frac{\partial X_i^k}{\partial \theta_m} = \frac{\partial Z_i^k}{\partial \theta_m}. \tag{3.39}$$

Since $\frac{\partial X_i^k}{\partial \theta_m}$ and $\frac{\partial Z_i^k}{\partial \theta_m}$ are linear function of $\frac{\partial t_i^k}{\partial \theta_m}$ as shown in the following equation:

$$\frac{\partial}{\partial \theta_m}\begin{bmatrix} X_i^k \\ Y_i^k \\ Z_i^k \end{bmatrix} = \frac{\partial R}{\partial \theta_m} \times \left(\tilde{p}_i^k t_i^k - O_f\right) + R \times \tilde{p}_i^k \frac{\partial t_i^k}{\partial \theta_m} \tag{3.40}$$

(which is the derivative of (3.17) with respect to θ_m), the derivatives of t_i^k with respect to rotation angles can be obtained.

That is,

$$\frac{\partial t_i^k}{\partial \theta_1} = -\frac{C - g'(X_i^k)A}{D - g'(X_i^k)B} \tag{3.41}$$

where

$$A = \left(\frac{\partial \hat{r}_1}{\partial \theta_1}\right)^T \tilde{p}_i^k t_i^k - \frac{\partial r_{13}}{\partial \theta_1} f \tag{3.42}$$

$$B = \hat{r}_1^T \tilde{p}_i^k \tag{3.43}$$

$$C = \left(\frac{\partial \hat{r}_3}{\partial \theta_1}\right)^T \tilde{p}_i^k t_i^k - \frac{\partial r_{33}}{\partial \theta_1} f \tag{3.44}$$

$$D = \hat{r}_3^T \tilde{p}_i^k. \tag{3.45}$$

$$\frac{\partial t_i^k}{\partial \theta_2} = -\frac{G - g'(X_i^k)E}{H - g'(X_i^k)F} \tag{3.46}$$

where

$$E = \left(\frac{\partial \hat{r}_1}{\partial \theta_2}\right)^T \tilde{p}_i^k t_i^k - \frac{\partial r_{13}}{\partial \theta_2} f \tag{3.47}$$

$$F = \hat{r}_1^T \tilde{p}_i^k \tag{3.48}$$

$$G = \left(\frac{\partial \hat{r}_3}{\partial \theta_2}\right)^T \tilde{p}_i^k t_i^k - \frac{\partial r_{33}}{\partial \theta_2} f \tag{3.49}$$

$$H = \hat{r}_3^T \tilde{p}_i^k. \tag{3.50}$$

$$\frac{\partial t_i^k}{\partial \theta_3} = -\frac{K - g'(X_i^k) I}{L - g'(X_i^k) H} \tag{3.51}$$

where

$$I = \left(\frac{\partial \hat{r}_1}{\partial \theta_3}\right)^T \tilde{p}_i^k t_i^k - \frac{\partial r_{13}}{\partial \theta_3} f \tag{3.52}$$

$$J = \hat{r}_1^T \tilde{p}_i^k \tag{3.53}$$

$$K = \left(\frac{\partial \hat{r}_3}{\partial \theta_3}\right)^T \tilde{p}_i^k t_i^k - \frac{\partial r_{33}}{\partial \theta_3} f \tag{3.54}$$

$$L = \hat{r}_3^T \tilde{p}_i^k. \tag{3.55}$$

Derivatives of t_i^k with respect to surface coefficients

In order to compute $\frac{\partial t_i^k}{\partial a_m}$, both sides of (3.16) also need to be differentiated with respect to a_m:

$$\frac{\partial g(X_i^k)}{\partial a_m} = \frac{\partial Z_i^k}{\partial a_m} \tag{3.56}$$

$$\left(X_i^k\right)^m + g'\left(X_i^k\right)\frac{\partial X_i^k}{\partial a_m} = \frac{\partial Z_i^k}{\partial a_m} \tag{3.57}$$

$$\left(X_i^k\right)^m + g'\left(X_i^k\right)\hat{r}_1^T \tilde{p}_i^k \frac{\partial t_i^k}{\partial a_m} = \hat{r}_3^T \tilde{p}_i^k \frac{\partial t_i^k}{\partial a_m}. \tag{3.58}$$

Finally, it is derived as

$$\frac{\partial t_i^k}{\partial a_m} = \frac{\left(X_i^k\right)^m}{\hat{r}_3^T \tilde{p}_i^k - g'(X_i^k)\hat{r}_1^T \tilde{p}_i^k}. \tag{3.59}$$

To force the document surface to pass O_f, a_0 is fixed to 0 during the optimization. The cost function (3.18) is not convex and has several local minima. Therefore, minimization starts from several initial points having typical view points and GCS parameters are set to 0 (planes), and optimal parameters are found by selecting the smallest residual. In solving (3.16) and (3.17) to get t_i^k, although there are generally

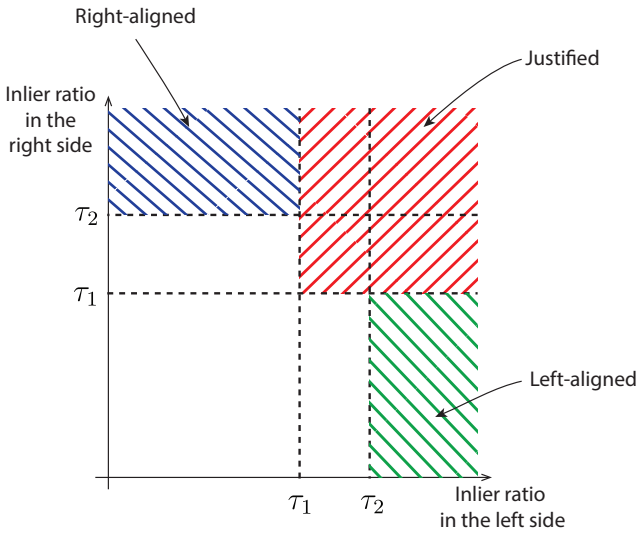


Figure 3.3: Decision boundaries for alignment type classification.

more than one solution, true one is obtained by using old t_i^k value (which were used in the previous iteration) as the initial value of Newton-Raphson method.

3.2.3 Alignment type classification

The minimization of $E_{str}(\cdot)$ provides rough rectification results and this facilitates the classification: text-lines are closed to horizontal straight lines and geometric distortion caused by page curls is almost removed in the results. However, the text-line extraction algorithm is not perfect: some text-lines are missing and broken as shown in Fig. 3.6-(a) and (e), therefore a robust algorithm is required. First, CCs of each text-line are sorted with respect to X_i^k, and then leftmost and rightmost points in the text-line are obtained. In the process, short text-lines (less than 80% of average text-line width in that text-block) are discarded to remove broken lines. Then, each side of remaining text-lines is fitted with the random sample consensus

(RANSAC) algorithm [15]. When the inlier ratios are high in either side of the text-block, the alignment types is employed in optimization. Otherwise, the cost function in $E_1(\cdot)$ is optimized. The proposed decision criterion is illustrated in Fig. 3.3. Since $E_1(\cdot)$ yields reasonable results in many situations, the decision whether to use $E_2(\cdot)$ is basically conservative (i.e., high τ_1 and τ_2). Moreover, in the justified text-blocks, the intersection between two side-lines is distant from the center of the text-block (e.g., the distance between the intersection and the center should be longer than the height of an input image to ensure that two side lines are parallel). Thus, the justified text-blocks which violate this condition are omitted in the optimization.

3.2.4 Design of $E_{align}(\cdot)$

Without loss of generality, it can be assumed that

$$X_1^k < X_2^k < \cdots \tag{3.60}$$

and the cost function for a left-aligned text-block is given by

$$E_{align}(\cdot) = \lambda_1 \sum_k \left(X_1^k - l \right)^2 \tag{3.61}$$

where l is the position of the left-alignment. Cost function for right-aligned text-block is similarly defined. For justified blocks, both cost functions are used. The derivatives of $\left(X_1^k - l\right)$ can be derived in a similar way to the previous section, and we skip the derivations.

Algorithm 1 Estimation of the alignment type in the text-block

Input:

I_L, the ratios of inliers to the text lines in left text,

I_R, the ratios of inliers to the text lines in right text,

P_v, vanishing point of two estimated side lines,

P_c, center point of the paragraph.

Parameter:

τ_l, threshold to ensure reliability in line estimation,

τ_p, threshold to decide whether two side lines are parallel or not,

τ_1, thresholds for alignment,

τ_2, thresholds for alignment, $\tau_2 > \tau_1$.

```
if I_L ≥ τ_l and I_R ≥ τ_l then
    if Distance(P_v, P_c) ≤ τ_p then
        if I_L ≥ τ_1 and I_R ≥ τ_1 then
            return justified
        else if I_L ≥ τ_1 then
            return left-aligned
        else if I_R ≥ τ_1 then
            return right-aligned
        else
            return center-aligned
        end if
    else
        return center-aligned
    end if
else if I_L ≥ τ_2 then
    return left-aligned
else if I_R ≥ τ_2 then
    return right-aligned
else
    return center-aligned
end if
```

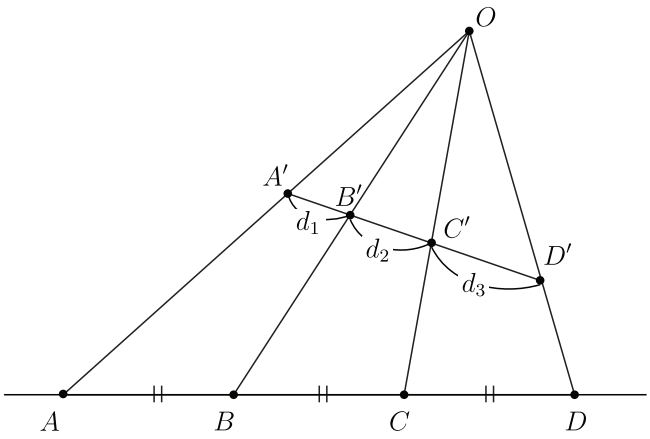


Figure 3.4: Projective transformation between two planar surfaces.

3.2.5 Design of $E_{spacing}(\cdot)$

In most cases, documents have the fixed line-spacings in text-blocks, and $E_{spacing}(\cdot)$ encodes this constraint. This term is basically given by

$$E_{spacing}(\cdot) = \lambda_2 \sum_k \left(l^{k-1} - 2l^k + l^{k+1} \right)^2 \tag{3.62}$$

where l^{k-1}, l^k, and l^{k+1} are vertical positions of consecutive text-lines within a text-block. Before optimization, in order to deal with outliers, text-lines within a text-block are sorted with respect to their vertical positions and outliers (which occur due to missing or broken text-lines) are removed.

Since geometric distortion caused by page curl is almost removed in the rough rectification with (3.11), the image acquisition model in the resultant image is equivalent to the projective transformation as shown in Fig. 3.4. Under projective transformation, the cross ratio of four collinear points is invariant, which is defined by

$$\mathrm{Cross}(A, B, C, D) = \frac{(A - B) \cdot (C - D)}{(A - C) \cdot (B - D)}. \tag{3.63}$$

In Fig. 3.4, if four points A, B, C, and D are located with equal spacing, cross ratio of the points is $\frac{1}{4}$ and cross ratio of transformed points (i.e., A', B', C', and D') has the same value. The equation of cross ratio of the transformed points is given by

$$\mathrm{Cross}\big(A', B', C', D'\big) = \frac{d_1}{(d_2 + d_3)} \cdot \frac{d_3}{(d_1 + d_2)} = \frac{1}{4}, \tag{3.64}$$

where d_1, d_2 and d_3 are distances between adjacent points as shown in Fig. 3.4. The equation about line spacing in the rough rectification result is derived from (3.64), which is given by

$$d^{k+1} = \frac{d^k \left(d^{k-1} + d^k\right)}{3d^{k-1} - d^k}, \tag{3.65}$$

where d^k is the k-th line spacing and defined as $d^k = l^{k+1} - l^k$.

Under this model, RANSAC algorithm is performed to get inliers and only inliers are involved in (3.62). To be specific, three consecutive text-lines within a text-block are randomly chosen, from which vertical positions of other text-lines are estimated using (3.65). Then, the position of each text-lines is compared with the estimated position, and the number of inliers are counted. This process is repeated and a set containing the maximum number of inliers is chosen.

Since the proposed cost function is additive, $E_1(\cdot)$ and $E_2(\cdot)$ can be minimized with the Levenberg-Marquardt algorithm.

3.3 Extension to unfolded book surfaces

In this section, the proposed approach is extended to unfolded book surfaces as shown in Fig. 3.8. For the goal, two GCS models are adopted for left and right surfaces respectively under the same cost function as the single page algorithm, and

the surface function is given by

$$g^+(x) = \begin{cases} g_l(x-T) & \text{if } x < T \\ 0 & \text{if } x = T \\ g_r(x-T) & \text{if } x > T \end{cases} \tag{3.66}$$

where $g_l(x) = \sum a_m^l x^m$, $g_r(x) = \sum a_m^r x^m$ and T is a new parameter representing the horizontal position of the book binding. In order to make $g^+(x)$ continuous, a_0^l and a_0^r are set to 0. Given a surface model,

$$g^+\left(X_i^k\right) = Z_i^k. \tag{3.67}$$

and (3.17) have to be solved in order to get the corresponding point of p_i^k on the surface. However, $g^+(x)$ is a piecewise polynomial and it is difficult to compute the intersection directly. In order to deal with this piecewise model, intersections with left and right surfaces are computed respectively and a proper one is chosen. To be precise, the intersection between the ray and the left surface is calculated by solving

$$g_l\left(X_{i,l}^k - T\right) = Z_{i,l}^k \tag{3.68}$$

where

$$X_{i,l}^k = \hat{r}_1^T\left(\tilde{p}_i^k t_{i,l}^k - O_f\right) \tag{3.69}$$

$$Z_{i,l}^k = \hat{r}_3^T\left(\tilde{p}_i^k t_{i,l}^k - O_f\right) \tag{3.70}$$

and $\tilde{p}_i^k t_{i,l}^k$ is the intersection point. Similarly $t_{i,r}^k$ and $X_{i,r}^k$ are obtained by computing an intersection with the right surface. When $(t_{i,l}^k, X_{i,l}^k)$ is a correct pair for (t_i^k, X_i^k), $X_{i,l}^k$ should belong to the left page and the ray from $\tilde{p}_i^k$ has to meet the left page first. Based on these observations (including the other case), following conditions are derived:

- If $X_{i,l}^k \leq T$ and $|t_{i,l}^k| \leq |t_{i,r}^k|$, then $t_i^k = t_{i,l}^k$,
- If $X_{i,r}^k \geq T$ and $|t_{i,l}^k| \geq |t_{i,r}^k|$, then $t_i^k = t_{i,r}^k$.

When neither candidate satisfies the conditions, the sample is skipped in the summation of cost function. In case that $t_{i,l}^k$ is chosen, the document surface is set to $g(x) = g_l(x - T)$ and the derivatives are derived like the single page case. Also in case that $t_{i,r}^k$ is chosen, the derivatives are derived in a similar way. The derivative of the residual (3.19) with respect to T is given by

$$\frac{\partial E(k,i)}{\partial T} = \hat{r}_2^T \tilde{p}_i^k \frac{\partial t_i^k}{\partial T}. \tag{3.71}$$

For the computation of $\frac{\partial t_i^k}{\partial T}$, both sides of (3.16) are differentiated with respect to T:

$$\frac{\partial g\big(X_i^k\big)}{\partial T} = \frac{\partial Z_i^k}{\partial T} \tag{3.72}$$

$$g_*{}'\Big(X_i^k - T\Big)\left(\frac{\partial X_i^k}{\partial T} - 1\right) = \frac{\partial Z_i^k}{\partial T} \tag{3.73}$$

$$g_*{}'\Big(X_i^k - T\Big)\left(\hat{r}_1^T \tilde{p}_i^k \frac{\partial t_i^k}{\partial T} - 1\right) = \hat{r}_3^T \tilde{p}_i^k \frac{\partial t_i^k}{\partial T}. \tag{3.74}$$

and final equation is derived as

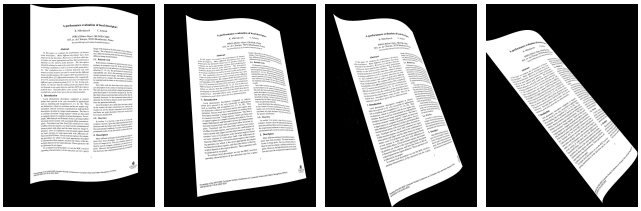
$$\frac{\partial t_i^k}{\partial T} = \frac{g_*{}'\big(X_i^k - T\big)}{\hat{r}_3^T \tilde{p}_i^k - g'\big(X_i^k - T\big)\hat{r}_1^T \tilde{p}_i^k}, \tag{3.75}$$

where $g_*(x)$ is the selected GCS equation between g_l and g_r.

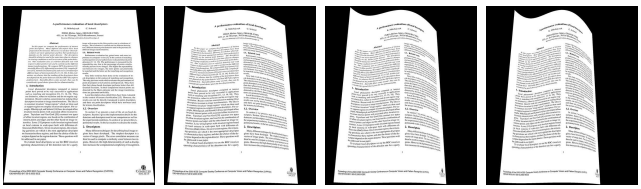
3.4 Experimental result

In this section, the proposed method is evaluated on synthetic and real images. For all the experiments, given focal lengths are used. In the implementation, the surface polynomial is given by

$$g(x) = \frac{1}{\sigma} h(\sigma x) \tag{3.76}$$



(a) $\|\Theta\| = 0^\circ$ (b) $\|\Theta\| = 20^\circ$ (c) $\|\Theta\| = 40^\circ$ (d) $\|\Theta\| = 60^\circ$



(e) $c = 0$ (f) $c = 1$ (g) $c = 2$ (h) $c = 3$

Figure 3.5: Examples of synthetic images.

where

$$h(x) = \sum_{m=0}^{M} a_m x^m, \tag{3.77}$$

and σ is used to deal with scale issues (numerical stability). The parameters are set as $M = 4$, $\sigma = 1000$, $\tau_1 = 0.4$, and $\tau_2 = 0.6$ for all experiments. Non-optimized C++ implementation of the proposed algorithm (only a single thread is used) takes 7–10 seconds to process an image (3264×2448) on Intel(R) i5(TM) CPU (3.2GHz). Most of time is spent on text-line extraction, which takes 4–5 seconds [3]. Optimization time depends on initial values for distortion parameters and the number of text-lines, and takes 1–4 seconds. Rendering the rectified image takes about 1 second. The proposed algorithm is publicly available at the website (`http://ispl.snu.ac.kr/bskim/DocumentDewarping/`), where input images and more results can be found.

Table 3.1: Mean rectification error with respect to the norm of rotation vector. Errors are measured in the unit of pixel and the values in the parentheses are the standard deviations.

Test image	0°	10°	20°	30°	40°	50°	60°
No.1	0.06(0.03)	0.06(0.05)	0.18(0.10)	0.17(0.10)	0.31(0.18)	0.97(0.61)	79.26(48.57)
No.2	0.07(0.03)	0.09(0.04)	0.27(0.14)	0.26(0.15)	0.36(0.17)	0.60(0.33)	0.70(0.40)
No.3	0.04(0.03)	0.05(0.03)	0.02(0.01)	0.04(0.02)	0.08(0.05)	50.49(32.06)	85.56(54.17)
No.4	0.05(0.03)	0.02(0.01)	0.02(0.01)	0.04(0.02)	0.06(0.03)	0.09(0.05)	0.36(0.22)
No.5	0.04(0.03)	0.05(0.03)	0.10(0.06)	0.11(0.06)	0.16(0.08)	0.16(0.09)	0.23(0.15)
No.6	0.23(0.13)	0.07(0.04)	0.09(0.05)	0.41(0.20)	2.31(1.32)	20.28(11.92)	28.69(18.01)
No.7	0.09(0.05)	0.07(0.04)	0.13(0.08)	0.17(0.09)	0.18(0.10)	0.19(0.11)	0.23(0.13)
No.8	0.01(0.01)	0.02(0.01)	1.33(0.70)	0.92(0.48)	2.92(1.69)	16.42(9.73)	30.53(17.91)
No.9	0.04(0.02)	0.08(0.04)	0.09(0.04)	0.12(0.07)	0.17(0.11)	0.26(0.14)	0.41(0.21)
No.10	0.02(0.01)	0.02(0.01)	0.01(0.01)	0.01(0.01)	0.10(0.05)	0.07(0.04)	3.03(1.98)
No.11	0.07(0.05)	0.10(0.07)	0.06(0.04)	0.15(0.08)	0.23(0.13)	0.18(0.12)	0.34(0.18)
No.12	0.02(0.01)	0.01(0.01)	0.02(0.01)	0.01(0.01)	0.15(0.09)	0.15(0.09)	0.23(0.10)
Average	0.06(0.04)	0.05(0.03)	0.19(0.11)	0.20(0.11)	0.58(0.33)	7.49(4.60)	19.13(11.84)

3.4.1 Experiments on synthetic data

First, experiments on simulated warping are conducted for the objective evaluations. As shown in Fig. **??**, 12 reference images with various properties such as two columns, text with tables/pictures, small number of text-lines are selected, and camera-captured images are synthesized as shown in Fig. 3.5. In order to focus on the performance of rectification algorithm, the text-lines are extracted from the scanned images and their locations in the simulated images are computed with the ground-truth distortion parameters.

From the ground-truth distortion parameters, the true dewarping functions are

Table 3.2: Mean rectification error with respect to surface complexities. Errors are measured in the unit of pixel and the values in the parentheses are the standard deviations.

Test image	$c = 0$	$c = 0.5$	$c = 1$	$c = 1.5$	$c = 2$	$c = 2.5$	$c = 3$
No.1	0.29(0.14)	0.08(0.04)	0.18(0.09)	0.12(0.07)	0.25(0.13)	0.65(0.35)	0.61(0.28)
No.2	1.85(0.95)	0.05(0.04)	0.12(0.08)	0.17(0.13)	0.22(0.15)	0.32(0.15)	0.43(0.22)
No.3	0.60(0.31)	0.02(0.01)	0.02(0.01)	0.02(0.01)	0.56(0.30)	0.66(0.31)	0.51(0.28)
No.4	0.78(0.36)	0.01(0.01)	0.02(0.01)	0.02(0.01)	0.02(0.01)	0.03(0.02)	0.02(0.01)
No.5	0.63(0.32)	0.04(0.03)	0.06(0.04)	0.09(0.05)	0.18(0.13)	0.65(0.30)	0.67(0.30)
No.6	1.11(0.52)	0.17(0.10)	0.01(0.01)	0.01(0.01)	0.01(0.01)	76.95(45.28)	15.20(9.56)
No.7	2.20(1.28)	0.03(0.02)	0.06(0.03)	0.12(0.08)	0.19(0.10)	0.24(0.12)	0.51(0.27)
No.8	2.04(1.11)	2.10(1.13)	0.02(0.01)	0.73(0.36)	4.62(2.70)	0.67(0.32)	51.14(32.35)
No.9	7.15(4.77)	0.05(0.04)	0.08(0.05)	0.14(0.07)	0.49(0.24)	0.20(0.13)	0.43(0.23)
No.10	2.83(1.74)	0.01(0.01)	0.01(0.01)	0.01(0.01)	0.01(0.01)	0.01(0.01)	0.59(0.30)
No.11	1.93(1.20)	0.03(0.02)	0.06(0.04)	0.11(0.05)	0.11(0.07)	0.19(0.10)	0.33(0.15)
No.12	4.05(2.22)	0.04(0.03)	0.01(0.01)	0.01(0.01)	0.05(0.03)	0.59(0.27)	0.61(0.28)
Average	2.12(1.24)	0.22(0.12)	0.05(0.03)	0.13(0.07)	0.56(0.32)	6.76(3.95)	5.92(3.69)

first computed in (**??**), and the function are compared with the estimated one. The distance metric is given by

$$E_r = \min_{\mu\in\Re,\nu\in\Re^2} \sqrt{\frac{1}{|\mathcal{P}|}\sum_{(\alpha,\beta)\in\mathcal{P}} |T_{gt}(\alpha,\beta) - \mu T(\alpha,\beta) - \nu|^2} \tag{3.78}$$

where $\mathcal{P}$ is a set of the points on the central part of document image, $T_{gt}(\cdot)$ is the ground truth dewarping function, and $T(\cdot)$ is the estimated function. Since the rectified results are equivalent up to the translation and scaling, $\mu \in \Re$ and $\nu \in \Re^2$ showing the minimum root mean square errors (RMSE) are selected.

Table 3.1 shows the RMSE with respect to the norm of rotation vector Θ. The norm of rotation vector means its rotation angle, and this experiment shows the

Table 3.3: Mean rectification error with respect to noise in text-lines. Errors are measured in the unit of pixel and the values in the parentheses are the standard deviations.

Test image	$\sigma = 0$	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	$\sigma = 4$	$\sigma = 5$	$\sigma = 6$
No.1	0.20(0.10)	0.21(0.12)	0.38(0.22)	0.18(0.11)	0.26(0.13)	0.88(0.44)	0.68(0.35)
No.2	0.22(0.12)	0.40(0.23)	0.50(0.26)	0.87(0.39)	1.75(0.93)	0.69(0.35)	2.97(1.59)
No.3	0.02(0.01)	0.11(0.05)	0.54(0.31)	0.45(0.28)	0.35(0.19)	1.77(0.93)	0.78(0.38)
No.4	0.02(0.01)	0.20(0.10)	0.38(0.22)	1.45(0.71)	0.70(0.32)	3.22(1.78)	0.97(0.48)
No.5	0.10(0.05)	0.21(0.13)	0.25(0.16)	0.25(0.15)	0.62(0.32)	0.58(0.31)	0.72(0.31)
No.6	0.11(0.06)	0.64(0.34)	2.01(1.02)	5.96(3.41)	8.75(5.38)	6.59(3.89)	12.06(7.22)
No.7	0.08(0.06)	0.10(0.07)	0.36(0.18)	0.85(0.37)	0.78(0.34)	0.84(0.38)	1.27(0.61)
No.8	1.37(0.73)	1.58(0.87)	5.37(3.05)	11.01(6.19)	10.79(6.23)	19.99(11.51)	21.37(12.56)
No.9	0.08(0.06)	0.17(0.11)	0.59(0.34)	1.18(0.66)	0.31(0.20)	2.92(1.62)	3.18(1.82)
No.10	0.01(0.01)	0.46(0.24)	0.97(0.47)	1.64(0.90)	5.84(3.43)	5.38(3.15)	5.35(3.21)
No.11	0.06(0.04)	0.19(0.13)	0.19(0.12)	0.53(0.29)	0.39(0.22)	0.36(0.19)	0.54(0.24)
No.12	0.02(0.01)	0.16(0.10)	0.28(0.16)	0.39(0.23)	0.52(0.29)	0.70(0.31)	0.93(0.43)
Average	0.19(0.11)	0.37(0.20)	0.99(0.54)	2.06(1.14)	2.59(1.50)	3.66(2.07)	4.23(2.43)

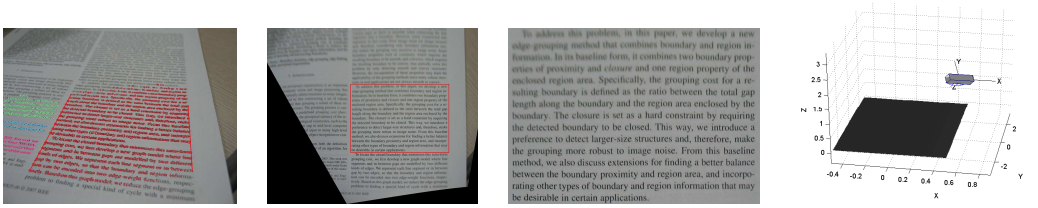
robustness of the proposed algorithm to the shot angle. As shown in Table 3.1, the proposed algorithm removes geometric distortions in the camera-captured document images for a range of rotation angles. For large angles (above 50°), the proposed algorithm sometimes fails due to the local minima. However, in most cases, the errors are less than 1 pixel and their standard deviations are also below 1 pixel.

For the evaluation of the proposed algorithm for a range of surfaces, a typical viewpoint $\Theta = \{-0.4, 0, -0.1\}$ is chosen and polynomial coefficients are set to $\{a_i\}_{i=0}^{4} = \{0, 0, 0.8, -0.7, -1.0\}$, and the surface complexities are increased by multiplying constants $c \in [0, 3]$ to the coefficients as shown in Fig. 3.5. The results of these experiments are summarized in Table 3.2, which shows that the proposed

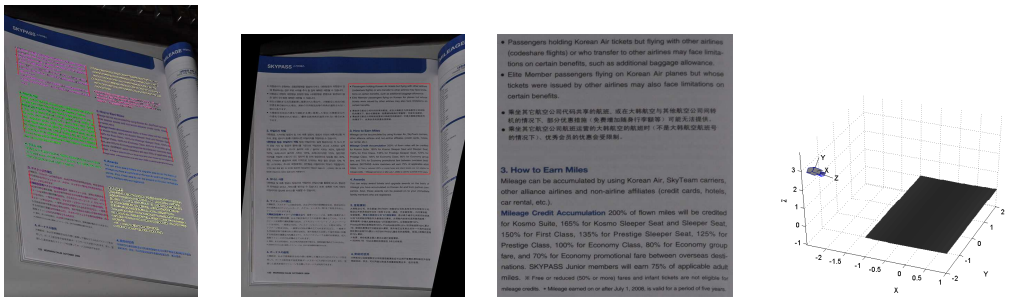
algorithm works for a range of geometric distortions. Interestingly, the proposed algorithm shows relatively large RMSE for the planar cases (i.e., $c = 0$), which seems to be an easy case. Actually, it is caused by the ambiguity in (3.11) for planar documents and the result demonstrates that the rectification of planar surfaces is not a simple task compared with the curved ones. Finally, the proposed algorithm is evaluated in the presence of errors in text-line extraction results, by adding white Gaussian noise in the CC positions. As shown in Table 3.3, the proposed method is robust to the errors in text-line extraction methods.

3.4.2 Experiments on real images

For real data, focal length information is obtained from the exchangeable image file format (EXIF) tag, which is available in most digital cameras. Although this number is not very accurate [26], the proposed method works well with them. Rectification results on planar and curved surfaces are shown in Fig. 3.6 and Fig. 3.7. As shown, the proposed method reduces nonlinear distortions as well as perspective distortions without the knowledge on surface type. Also, note that the proposed method works for complex layouts, and it is robust to non-textual objects and background clutters. Rectification results on unfolded book surfaces are shown in Fig. 3.8, where red lines indicate book bindings. Although the number of parameters increases due to book surface models, the proposed method successfully works. Moreover, the reconstructed document surfaces using estimated parameters are shown in the last columns of Figs. 3.6–3.8, and these also validate correctness of the proposed method.

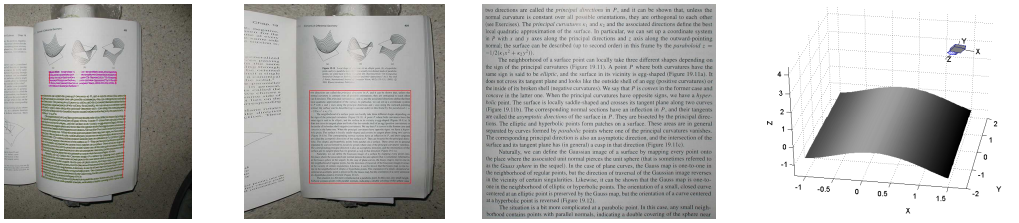


(a) Input image (b) Rectified image (c) Enlarged subimage (d) Reconstructed surface

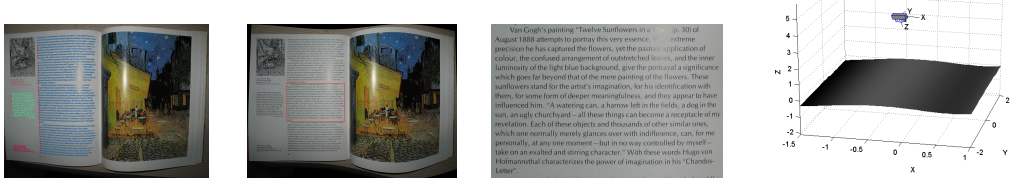


(e) Input image (f) Rectified image (g) Enlarged subimage (h) Reconstructed surface

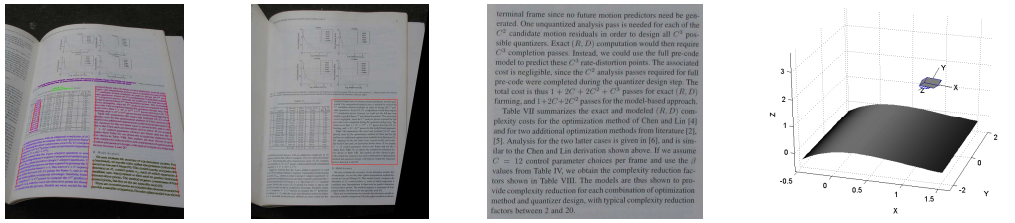
Figure 3.6: Rectification results of planar documents. Images in the first column are inputs, where colored circles are CCs in detected text-lines and red lines show the sides of each text-block. Images in the second column are the rectified images and regions inside red rectangles are enlarged in the third column. Figures in the last column show reconstructed document surfaces and cameras.



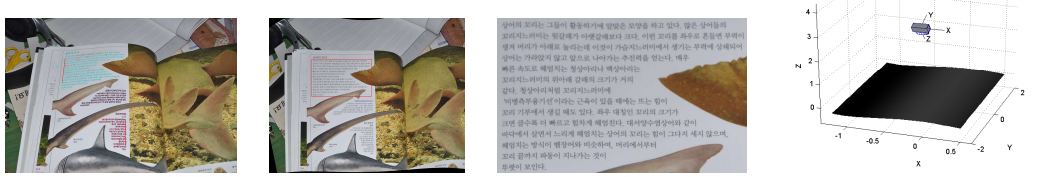
(a) Input image (b) Rectified image (c) Enlarged subimage (d) Reconstructed surface



(e) Input image (f) Rectified image (g) Enlarged subimage (h) Reconstructed surface

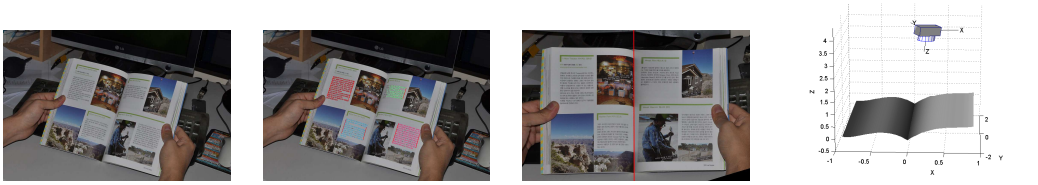


(i) Input image (j) Rectified image (k) Enlarged subimage (l) Reconstructed surface

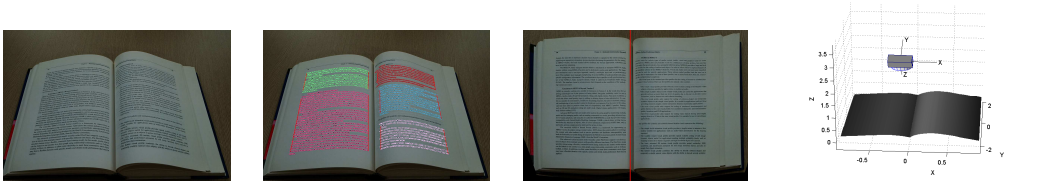


(m) Input image (n) Rectified image (o) Enlarged subimage (p) Reconstructed surface

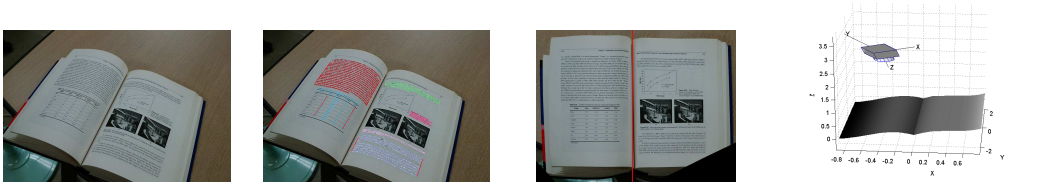
Figure 3.7: Rectification results of curved documents. Images in the first column are inputs, where colored circles are CCs in detected text-lines and red lines show the sides of each text-block. Images in the second column are the rectified images and regions inside red rectangles are enlarged in the third column. Figures in the last column show reconstructed document surfaces and cameras.



(a) Input image (b) Extracted text-lines (c) Rectified image (d) Reconstructed surface



(e) Input image (f) Extracted text-lines (g) Rectified image (h) Reconstructed surface



(i) Input image (j) Extracted text-lines (k) Rectified image (l) Reconstructed surface

(m) Input image (n) Extracted text-lines (o) Rectified image (p) Reconstructed surface

Figure 3.8: Rectification results of book images. Images in the first column are inputs, and images in the second column are text-line extraction results where colored circles are CCs in detected text-lines and red lines show the sides of each text-block. Images in the third column are the rectified images where the red lines indicate book bindings. Figures in the last column show reconstructed document surfaces and cameras.

3.4.3 Comparison with existing methods

For objective evaluation, the proposed algorithm is tested on the CBDAR2007 dewarping contest dataset [47] and compared with other algorithms.[1] The CBDAR2007 dataset includes camera-captured document images, and also the dewarping results of them obtained by five methods, i.e., SKEL [48], SEG [49], CTM [11], CTM2 ([11] followed by removing non-text regions from the images), and snakes [50]. However, the proposed method requires focal lengths which are not available in the dataset. To estimate the focal length, the dataset is rectified on a range of focal length value (from 30° FOV to 74° FOV), and a result showing the minimum residual is selected among them. The rectification results of the proposed method and other algorithms on several images in the dataset are shown in Fig. 3.9.

Since the SEG method use the local transform, the result of the SEG method shows uneven text lines. Although the SKEL method removes the geometric distortion such as the page curl, the perspective distortion still remains in the results. The CTM method removes the perspective and the geometric distortions, but it removes some text part near the left and right border due to failure in estimating boundary of text region. While the snake method also removes the perspective and the geometric distortions, non-texture region such as graphics is not rectified well. The proposed method corrects both perspective and geometric distortions, and shows good qualities in the non-texture regions.

For objective comparison, the rectification performance is evaluated by measuring the remaining geometric distortions in result images. However, the error metric in (3.78) cannot be applied directly, since the true dewarping function $T_{gt}(\cdot)$ is not

[1] The CBDAR2007 dataset can be downloaded from `http://www.csse.uwa.edu.au/~shafait/downloads.html`.

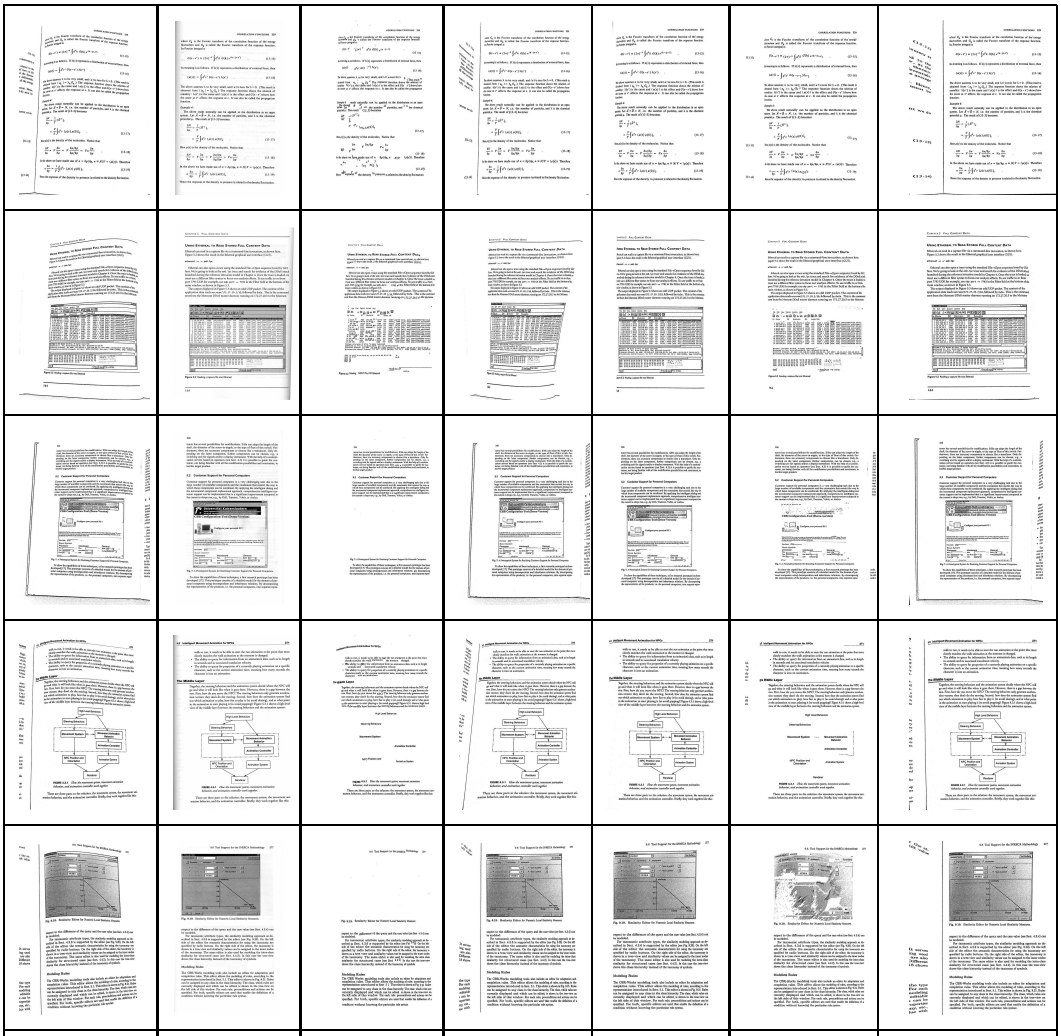


Figure 3.9: Dewarping results on CBDAR2007 data set. From left to right column, input image, scanned ground truth and the results obtained by SEG, SKEL, CTM, snakes, and the proposed method are shown.

available in this dataset. Therefore, similar to the evaluation method in [4], the error metric in (3.78) is evaluated by computing correspondences between scanned images and rectified ones. Fig. 3.10 shows the box plot of the errors in the rectified images,

where the "MRCDI" is the algorithm proposed in [4]. In the figure, the bottom and top of boxes indicate the first and third quartile, and red lines indicate the median values. Then, the band inside the box includes middle 50% of the data. Two lines extending vertically from the boxes, called whiskers, represent the rest of data, and the ends of whiskers indicate the lowest datum within 1.5 interquartile range of the lower quartile and highest datum within 1.5 interquartile range of the upper quartile [51]. Blue circles show the outliers which are not included between the whiskers. As shown in Fig. 3.10, the proposed method yields the best performance. Although the proposed method has a couple of outliers (probably due to the local minima), note that the MRCDI method removes border noise manually, while the proposed method works in the presence of border noise.

OCR accuracy is also evaluated on the rectification results. The OCR is performed using a commercial software, ABBYY FineReader Pro 9.0, and measure the accuracy, which is given by

$$\text{Accuracy}\,(\%) = 100 \times \left(1 - \frac{\text{Edit Distance}}{\max\left(N_s, N_t\right)}\right), \tag{3.79}$$

where the edit distance [52] is the minimum number of insertion, deletion, and substitution to make two strings the same, and N_s and N_t are the numbers of characters of two strings. Fig. 3.11 shows the box plot of OCR accuracy of the rectified images. The results are similar to the case of rectification error, and the proposed method compares favorably with the MRCDI method.

3.4.4 Limitations

The proposed method is based on GCS assumptions and pin-hone camera models. Therefore, the violation on these assumptions may yield poor results. Moreover,

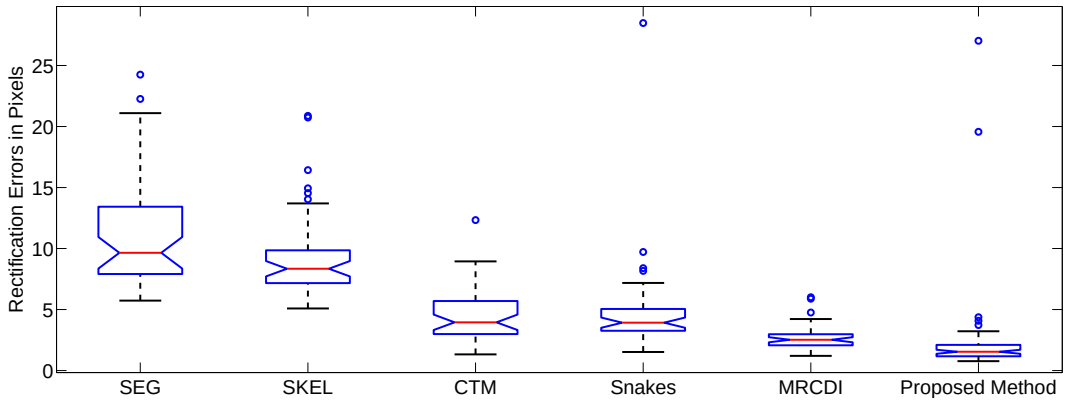


Figure 3.10: Box plot of the rectification error of six methods. The rectification errors of six methods are measured on CBDAR2007 data set (some results are from [4]).

surfaces near book bindings sometimes deviate from low-order polynomial models and they are not rectified well.

The proposed method assumes that the surface of a document is modeled by GCS and the pin-hole camera is used. If the case violating these assumptions occurs, the proposed algorithm does not work.

As a future work, while the proposed method focuses on correcting perspective and geometric distortion of the document image, removing photometric distortion such as shading effect is also required for good quality. Since the proposed method estimates shape of document surface, the surface normal can be computed from the estimated shape. Using the surface normal, shading can be removed and an albedo image can be estimated [13]. Besides, an extension of the GCS model to more general model is other possible topic. Currently the proposed algorithm can only rectify smooth surface, so the region nearby book binding or the folded document are not rectified well. To address the problem, employing piecewise polynomials to

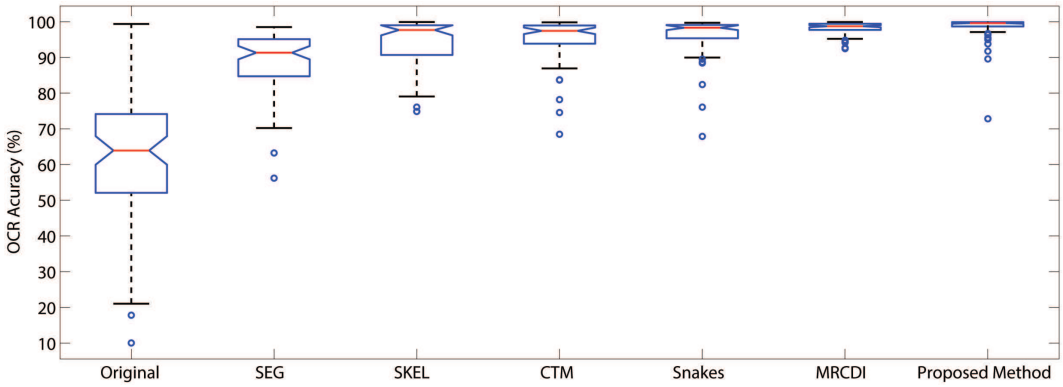


Figure 3.11: Box plot of the OCR accuracy of six methods. The OCR accuracies of six methods are measured on CBDAR2007 data set (some results are from [4]).

model the irregular surface can be an advantageous option.

Chapter 4

Document rectification based on feature detection

Conventional image stitching methods including document images have been developed assuming either of the fixed cameras or the planar target, and users have to know which condition is more appropriate. On the other hand, the proposed algorithm is able to handle both cases under the same framework as illustrated in Fig. 4.2 and 4.1. Also, the proposed approach provides a general image stitching framework for the planar documents (in a similar way as the conventional fixed-optical-center method [17]). As illustrated in Fig. 4.3, since the conventional document image stitching methods were based on pairwise registrations [31], they are susceptible to local minima and error propagations. On the other hand, the proposed cost function is based on the registration errors on the reference plane and thus alleviates such problems as shown in Fig. 4.3-(c).

Since the proposed method yields camera poses and rectified documents images when the camera motion is large, it can also be used for some interesting applications.

Note that the metric rectification of the documents images is very helpful in many applications such as optical character recognition (OCR) [53,54], document retrieval [55], and augmented reality (AR) [56]. In the conventional approaches, the metric rectification is achieved using some additional hardwares such as inertia sensors [56] or it was assumed that the user provides a fronto-parallel view at the first frame [57]. the proposed method addresses the rectification and stitching problem simultaneously, and therefore, it can be used in the applications such as camera-based document scanner and text-based AR.

4.1 Proposed approach

For the rectification of the camera-captured document image, most important step is to estimate the camera pose and the shape of the document. Motivated from image stitching and structure-from-motion [16,15], a novel approach to estimate the camera pose and the shape of the documents is proposed. Basically, the proposed method requires more than one image for rectification and use feature points instead of text-lines. The method consists of the detection of the corresponding points between images, the camera pose estimation by the optimizing the cost function, the determination of type of camera motion, and finally the tilt correction or the reference correction are performed depending on the type of camera motion.

The flowchart of the proposed method is shown in Fig. 4.2.

The pose of each camera is parameterized as an element in the special Euclidean group $SE(3)$ and a cost function whose minimization yields the pose is proposed. The cost function is defined on the reference plane (image stitching result) and it is minimized via the Levenberg-Marquardt algorithm [58]. From the estimated camera

poses, the motion model is determined whether the optical center is fixed or not: when the optical center is moved (i.e., the camera motion is large), metric rectification is possible and the proposed algorithm provides rectified composites as well as camera poses. Otherwise, the composite is built with respect to an appropriate view point.

The rest of the chapter is organized as follows. In Sec. 4.2, the proposed cost function and its minimization method for the camera pose estimation are presented. After the estimation, post-processing algorithms will be explained in Sec. 4.3. Experimental results on synthetic and real data are shown in Sec. 4.4.

4.2 Proposed cost function and its optimization

4.2.1 Notations

The internal matrix of the i-th camera and its pose (external parameters) are denoted by

$$\mathbf{K}_i = \begin{bmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.1}$$

and

$$\begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \in SE(3) \tag{4.2}$$

where $\mathbf{R}_i \in SO(3)$ is the rotation matrix and $\mathbf{t}_i = [t_{ix}, t_{iy}, t_{iz}]^T \in \Re^3$ is a translation vector. The rotation matrix is parameterized as the exponential representation:

$$\mathbf{R}_i = \exp\left([\boldsymbol{\theta}_i]_\times\right) \tag{4.3}$$

for a vector $\boldsymbol{\theta}_i = [\theta_{ix}, \theta_{iy}, \theta_{iz}]^T$. Finally, the reference plane is given by $\boldsymbol{\pi}_E = \left(\mathbf{n}^T, d\right)^T$. Without the loss of generality, we set $\mathbf{n} = [0, 0, 1]^T$ and $d = 0$, i.e., the reference plane is z-plane in the world coordinate system.

4.2.2 Homography between the i-th image and $\boldsymbol{\pi}_E$

The camera matrix of the i-th image [15] is given by

$$\mathbf{P}_i = \mathbf{K}_i \mathbf{R}_i \left[\mathbf{I}_{3\times 3}| - \mathbf{t}_i\right], \tag{4.4}$$

and the relationship between a point $[X, Y, 0]^T$ on the surface and its corresponding point on the i-th image $\mathbf{p} = [u, v]^T$ is given by

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}_i \mathbf{R}_i \left[\mathbf{I}_{3\times 3}| - \mathbf{t}_i\right] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} \tag{4.5}$$

$$= \mathbf{K}_i \begin{bmatrix} \mathbf{R}_i\mathbf{e}_1 & \mathbf{R}_i\mathbf{e}_2 & -\mathbf{R}_i\mathbf{t}_i \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{4.6}$$

for some λ. Thus, the homography $\mathbf{H}_i$ between the i-th image and the reference plane is

$$\mathbf{H}_i = \mathbf{K}_i \begin{bmatrix} \mathbf{R}_i\mathbf{e}_1 & \mathbf{R}_i\mathbf{e}_2 & -\mathbf{R}_i\mathbf{t}_i \end{bmatrix} \tag{4.7}$$

$$= \mathbf{K}_i \begin{bmatrix} \mathbf{R}_i\mathbf{e}_1 & \mathbf{R}_i\mathbf{e}_2 & \mathbf{R}_i\mathbf{e}_3 - \mathbf{R}_i\mathbf{t}_i - \mathbf{R}_i\mathbf{e}_3 \end{bmatrix} \tag{4.8}$$

$$= \mathbf{K}_i\mathbf{R}_i \left\{\mathbf{I} - (\mathbf{t}_i + \mathbf{e}_3)\, \mathbf{e}_3^T\right\} \tag{4.9}$$

where $\mathbf{e}_i$ $(i = 1, 2, 3)$ are the unit vectors representing each direction.

4.2.3 Proposed cost function

For the registration of a set of images, a new cost function reflecting the registration errors on a reference plane is developed. To be precise, given a correspondence $\mathbf{p}_i^k \leftrightarrow \mathbf{p}_j^m$ between the i-th image and j-th image ($\mathbf{p}_i^k$ denotes the k-th feature in the i-image), the registration error $\mathbf{r}_{i,j}^{k,m}$ is given by

$$\mathbf{r}_{i,j}^{k,m} = \mathbf{p}_{iX}^k - \mathbf{p}_{jX}^m \tag{4.10}$$

where $\mathbf{p}_{iX}^k$ and $\mathbf{p}_{jX}^m$ are the projected points of $\mathbf{p}_i^k$ and $\mathbf{p}_j^m$ respectively, to the reference plane as illustrated in Fig. 4.4. That is,

$$\tilde{\mathbf{p}}_{iX}^k = \mathbf{H}_i^{-1}\tilde{\mathbf{p}}_i^k \tag{4.11}$$

$$\tilde{\mathbf{p}}_{jX}^m = \mathbf{H}_j^{-1}\tilde{\mathbf{p}}_j^m \tag{4.12}$$

where the tilde is used for the homogeneous representation of points.

The proposed cost function is given by the sum of registration errors for all the correspondences

$$e = \sum_{i=1}^{N} \sum_{j \in I(i)} \sum_{k \in F(i,j)} \mathbf{r}_{i,j}^{k,m} \tag{4.13}$$

where N is the number of images, $I(i)$ is the set of images matched to the i-th image, and $F(i,j)$ is the set of correspondences between the i-th and j-th images. The correspondences are found by using the method in [17]: SIFT [59] features are extracted, correspondences are found by the nearest neighbor search, and inliers are selected using the random sample consensus (RANSAC) algorithm [60].

4.2.4 Optimization

Since the proposed cost function in (4.13) consists of sum of squared error, the Levenberg-Marquardt algorithm [58] is employed for the optimization. For the im-

plementation, the Jacobian of the registration error is derived analytically:

$$\frac{\partial \mathbf{r}_{i,j}^{k,m}}{\partial t} = \frac{\partial \mathbf{p}_{iX}^{k}}{\partial t} - \frac{\partial \mathbf{p}_{jX}^{m}}{\partial t}. \tag{4.14}$$

If t is a parameter related to the i-th image (i.e., $t \in \{\theta_{ix}, \theta_{iy}, \theta_{iz}, t_{ix}, t_{iy}, t_{iz}\}$), then the first term in the right-hand side can be derived as

$$\frac{\partial \mathbf{p}_{iX}^{k}}{\partial t} = \frac{\partial \mathbf{p}_{iX}^{k}}{\partial \tilde{\mathbf{p}}_{iX}^{k}} \frac{\partial \tilde{\mathbf{p}}_{iX}^{k}}{\partial t} \tag{4.15}$$

where

$$\frac{\partial \mathbf{p}_{iX}^{k}}{\partial \tilde{\mathbf{p}}_{iX}^{k}} = \frac{\partial \begin{bmatrix} x/z & y/z \end{bmatrix}}{\partial \begin{bmatrix} x & y & z \end{bmatrix}} = \begin{bmatrix} 1/z & 0 & -x/z^2 \\ 0 & 1/z & -y/z^2 \end{bmatrix}. \tag{4.16}$$

Since $\tilde{\mathbf{p}}_{iX}^{k} = \mathbf{H}_i^{-1}\tilde{\mathbf{p}}_i^{k}$ and $\mathbf{H}_i^{-1}$ can be obtained by applying the Matrix Inversion Lemma to (4.9):

$$\mathbf{H}_i^{-1} = \left\{ \mathbf{I} - \frac{(\mathbf{t}_i + \mathbf{e}_3)\,\mathbf{e}_3^T}{\mathbf{e}_3^T \mathbf{t}_i} \right\} \mathbf{R}_i^T \mathbf{K}_i^{-1}, \tag{4.17}$$

we can get $\frac{\partial \tilde{\mathbf{p}}_{iX}^{k}}{\partial t}$ for each parameter:

$$\frac{\partial \tilde{\mathbf{p}}_{iX}^{k}}{\partial \theta_{ix}} = \left\{ \mathbf{I} - \frac{(\mathbf{t}_i + \mathbf{e}_3)\,\mathbf{e}_3^T}{\mathbf{e}_3^T \mathbf{t}_i} \right\} \frac{\partial \mathbf{R}_i^T}{\partial \theta_{ix}} \mathbf{K}_i^{-1} \tilde{\mathbf{p}}_i^{k} \tag{4.18}$$

$$\frac{\partial \tilde{\mathbf{p}}_{iX}^{k}}{\partial \theta_{iy}} = \left\{ \mathbf{I} - \frac{(\mathbf{t}_i + \mathbf{e}_3)\,\mathbf{e}_3^T}{\mathbf{e}_3^T \mathbf{t}_i} \right\} \frac{\partial \mathbf{R}_i^T}{\partial \theta_{iy}} \mathbf{K}_i^{-1} \tilde{\mathbf{p}}_i^{k} \tag{4.19}$$

$$\frac{\partial \tilde{\mathbf{p}}_{iX}^{k}}{\partial \theta_{iz}} = \left\{ \mathbf{I} - \frac{(\mathbf{t}_i + \mathbf{e}_3)\,\mathbf{e}_3^T}{\mathbf{e}_3^T \mathbf{t}_i} \right\} \frac{\partial \mathbf{R}_i^T}{\partial \theta_{iz}} \mathbf{K}_i^{-1} \tilde{\mathbf{p}}_i^{k} \tag{4.20}$$

and

$$\frac{\partial \tilde{\mathbf{p}}_{iX}^k}{\partial t_{ix}} = -\frac{1}{t_{iz}} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{R}_i^T \mathbf{K}_i^{-1} \tilde{\mathbf{p}}_i^k \tag{4.21}$$

$$\frac{\partial \tilde{\mathbf{p}}_{iX}^k}{\partial t_{iy}} = -\frac{1}{t_{iz}} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{R}_i^T \mathbf{K}_i^{-1} \tilde{\mathbf{p}}_i^k \tag{4.22}$$

$$\frac{\partial \tilde{\mathbf{p}}_{iX}^k}{\partial t_{iz}} = \frac{1}{t_{iz}^2} \begin{bmatrix} 0 & 0 & t_{ix} \\ 0 & 0 & t_{iy} \\ 0 & 0 & 1 \end{bmatrix} \mathbf{R}_i^T \mathbf{K}_i^{-1} \tilde{\mathbf{p}}_i^k. \tag{4.23}$$

In the optimization, without the loss of generality, we assume that the first camera is on the z-axis, i.e., $t_{1x} = t_{1y} = 0$.

4.2.5 Relation to the model in [17]

From (4.9) and (4.17), pairwise homography between two views can be obtained:

$$\mathbf{H}_j \mathbf{H}_i^{-1} = \mathbf{K}_j \mathbf{R}_j \left\{ \mathbf{I} - \frac{(\mathbf{t}_i - \mathbf{t}_j)\, \mathbf{e}_3^T}{\mathbf{e}_3^T \mathbf{t}_i} \right\} \mathbf{R}_i^T \mathbf{K}_i^{-1}, \tag{4.24}$$

which reduces to

$$\mathbf{K}_j \mathbf{R}_j \mathbf{R}_i^T \mathbf{K}_i^{-1} \tag{4.25}$$

when the optical center is fixed (i.e., $\mathbf{t}_i = \mathbf{t}_j$), which is the same model used in [17].

4.3 Post-processing

By minimizing the cost function in (4.13), the location of each camera can be estimated. With the estimated camera position, the type of the camera motion is

determined. When the distance between the cameras is large (not fixed camera), the proposed method rectifies the images with the available structure information that can be obtained from the large camera motion. Conversely, when the camera motion is very small or the optical centers are fixed, the metric rectification is impossible and it would be better to provide stitching result without the metric rectification. In this section, the criterion on this decision and the proposed viewpoint selection method is presented.

4.3.1 Classification of two cases

From the estimated camera poses, the amount of camera motion is given by

$$\max_{1\leq i<j\leq N} \|\mathbf{t}_i - \mathbf{t}_j\|_2 \,. \tag{4.26}$$

Because (4.26) is proportional to the scale of the scene, the global scale is also estimated as:

$$\frac{1}{N}\sum_{i=1}^{N} |\mathbf{t}_i^T \mathbf{e}_3|, \tag{4.27}$$

which is the average distance between the cameras and the reference plane. Based on (4.26) and (4.27), the proposed criterion is given by

$$\Gamma = \frac{(4.26)}{(4.27)}. \tag{4.28}$$

When $\Gamma < \tau$ (i.e., camera motion is small compared with the scale of the scene), metric rectification is not reliable and a method in [61, 62] is applied. Otherwise (i.e., $\Gamma > \tau$), the metric rectification on the reference planes is performed.

4.3.2 Skew removal

For skew removal, a up-vector $\mathbf{u}$ is estimated based on the assumption that people do not severely twist the camera pose relative to the horizon when taking pictures

[17,63]. This process is illustrated in Fig. 4.5 and it is formulated as a minimization problem:

$$\mathbf{u} = \arg\min_{\|\mathbf{r}\|=1} \sum_{i=1}^{N} \left(\mathbf{r}^T \mathbf{r}_i\right)^2 \tag{4.29}$$

$$= \arg\min_{\|\mathbf{r}\|=1} \mathbf{r}^T \sum_{i=1}^{N} \left(\mathbf{r}_i \mathbf{r}_i^T\right) \mathbf{r} \tag{4.30}$$

where $\mathbf{r}_i$ is the first row of the i-th rotation matrix (i.e., $\mathbf{e}_1^T \mathbf{R}_i$). Therefore, $\mathbf{u}$ is the smallest eigenvector of the scatter matrix spanned by the x-directions of cameras as shown in Fig. 4.5-(c). Finally, the skews are removed by applying the global rotation

$$\mathbf{R}_g = \left[\left[\mathbf{u}'\right]_\times \mathbf{e}_3, \mathbf{u}', \mathbf{e}_3\right] \tag{4.31}$$

where $\mathbf{u}'$ is obtained by using the Gram-Schmidt orthonormalization.

4.4 Experimental results

In the experiments, the blending method in [17, 64] is applied for reducing the photometric errors. Since focal length is usually available in the exchangeable image file format (EXIF), this information is utilized in the experiments on real images.

4.4.1 Quantitative evaluation on metric reconstruction performance

For quantitative evaluation of the metric rectification performance of the proposed algorithm, synthetic pairs as shown in Fig. 4.6-(a) and (b) are built : each image is synthesized with randomly chosen viewpoint and focal length. Given a pair, the homography in (4.9) is estimated with the proposed algorithm and compared with the ground truth. The distance measure between the ground truth homography $\mathbf{H}_G$

and the estimated one $\hat{\mathbf{H}}_E$ is given by

$$D = \min_{\mathbf{\Lambda}} \sqrt{\frac{1}{|R|} \sum_{\mathbf{x} \in R} \left| g\left(\mathbf{H}_G^{-1} \tilde{\mathbf{x}}\right) - g\left(\mathbf{\Lambda} \hat{\mathbf{H}}_E^{-1} \tilde{\mathbf{x}}\right) \right|^2} \tag{4.32}$$

where R is a domain on which a given image is defined, $g(\cdot)$ converts a homogeneous vector to an inhomogeneous one (i.e., $g(\tilde{\mathbf{x}}) = \mathbf{x}$), and $\mathbf{\Lambda}$ is a similarity transform. Intuitively, the distance becomes small when the relationship between the estimated homography and the ground truth is given by a similarity transform. Equation (4.32) is computed with the method in [65].

The proposed method is evaluated on the 275 pairs of synthesis images. In order to evaluate the robustness to the errors in focal length, additional experiments is conducted by introducing the errors in focal length. To be specific, each pair is synthesized with focal length $f\left(1 + \frac{e_m}{100}\right)$ where $e_m \in \{\pm 5, \pm 3, \pm 1, 0\}$, and tried metric rectification with f. The results are summarized in Fig. 4.6-(c), which shows that the proposed algorithm yields almost perfect metric reconstruction performance with the true focal length values, and its performance decreases as $|e_m|$ increases. However, registration errors are less than 1.5 pixels for a range of errors ($-5\% \leq e_m \leq +5\%$).

4.4.2 Experiments on real images

When the camera center is fixed as shown in the first column of Fig. 4.1, the proposed method addresses the same problem in [17], but yields different results (See Sec.4.2.5). On Fig. 4.7-(a) shows six input images of fixed-optical-center cases. Fig. 4.7-(b) and (c) are document image stitching results using *Autostitch* [17] and the proposed method respectively. Since the proposed method adopts perspective projection to rendering the stitching result, it shows better visual qualities in planar

document images. On the contrary, *Autostitch* employs cylindrical projection and it results in the stitching image with bending distortion, which deforms straight lines in the scene into curved lines. In addition, the proposed method adopts a viewpoint selection algorithm in [61,62], the final composite shows little perceptual distortion. For considering the weakness of *Autostitch* in the case of plane-target, the proposed method is compared with a method based on the sequential registration using pairwise homographies. Fig. 4.8-(a) shows nine input images from a moving camera, and Fig. 4.8-(b) shows estimated camera poses with respect to the reference plane. Fig. 4.8-(c) and (d) are image stitching results using the conventional approach and the proposed method respectively. In order to highlight the perspective distortions, the red horizontal dotted lines are overlaid. As can be seen, the proposed method generates a fronto-parallel view successfully.

In addition to natural images, the proposed method can be applied to large documents image with dense text [30, 66]. As shown in Fig. 4.9, the proposed method yields rectified results without any text-specific information. Another example (without blending) can be found in Fig. 4.3-(c). The proposed method minimizes the global registration error and it less suffers from mis-registration by considering the all pairs of correspondences simultaneously. However, the conventional method takes corresponding points and images into account sequentially, the registration error is accumulated and it shows results deviated from rectified ones.

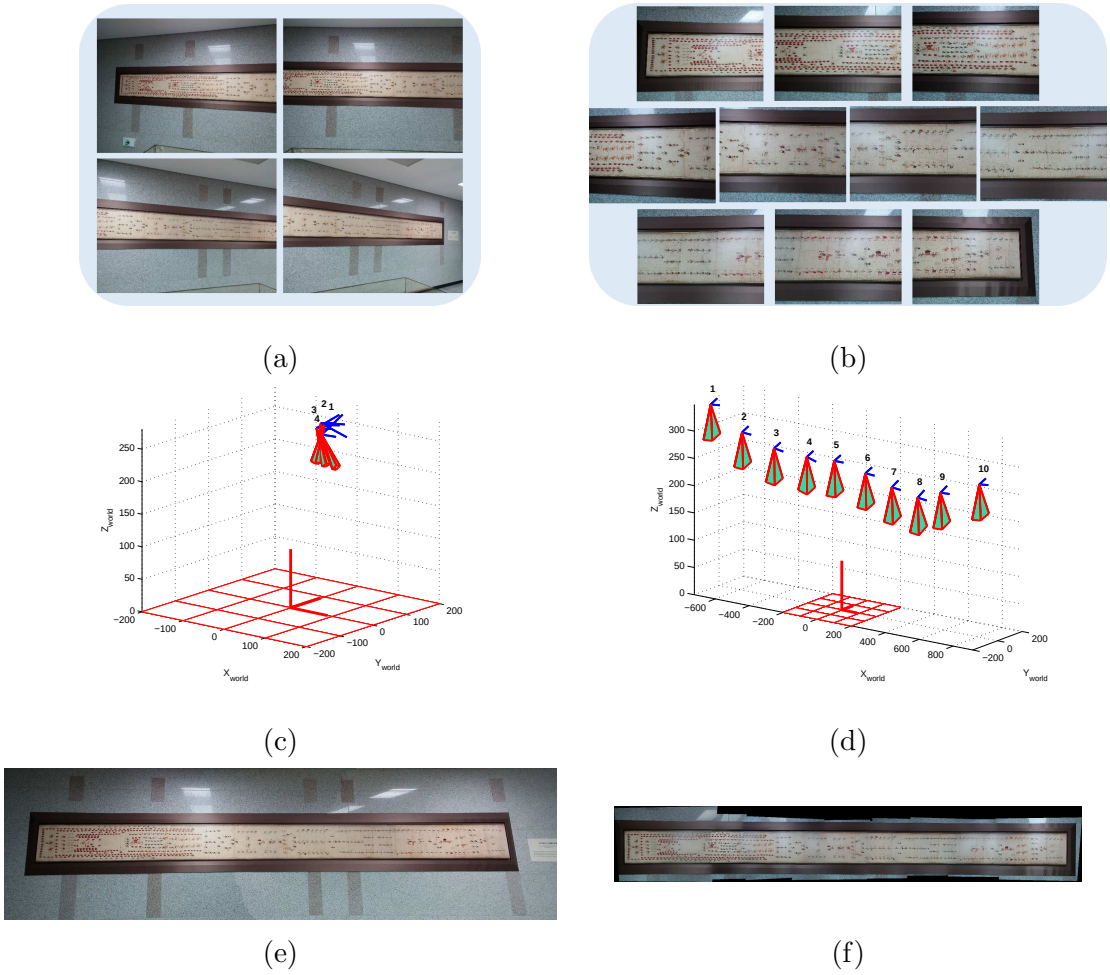


Figure 4.1: Results of the proposed algorithm. (a), (b) Input images, (c), (d) Estimated camera poses, (e), (f) Final results.

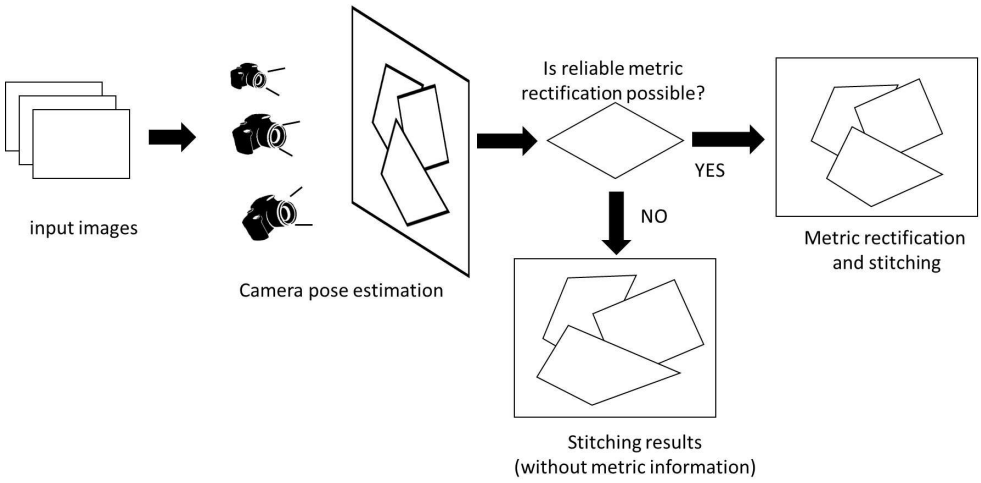
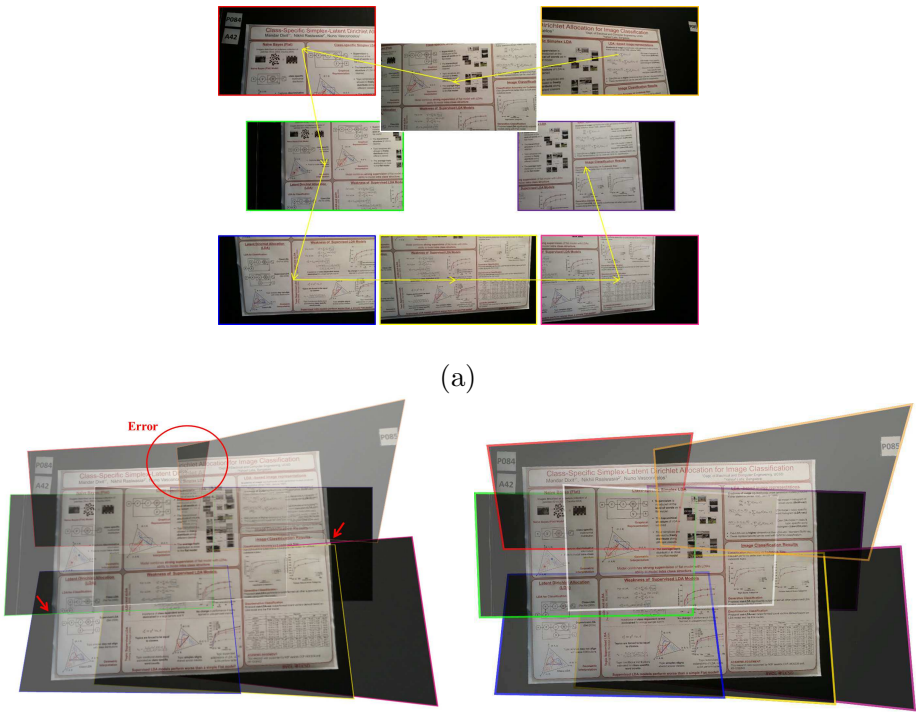


Figure 4.2: Flowchart of the proposed algorithm.



(a)

(b)

(c)

Figure 4.3: Comparison between the sequential registrations and the proposed method. (a) Eight input images capturing the same plane, (b) Document stitching result using the pairwise homographies. Note that the composite suffers from error accumulations, (c) Document stitching result using the proposed method. The proposed method minimizes the global registration errors and it less suffers from mis-registrations.

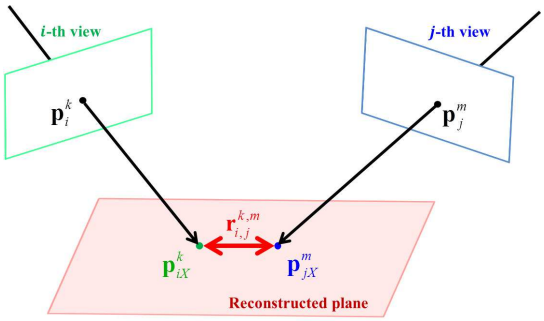
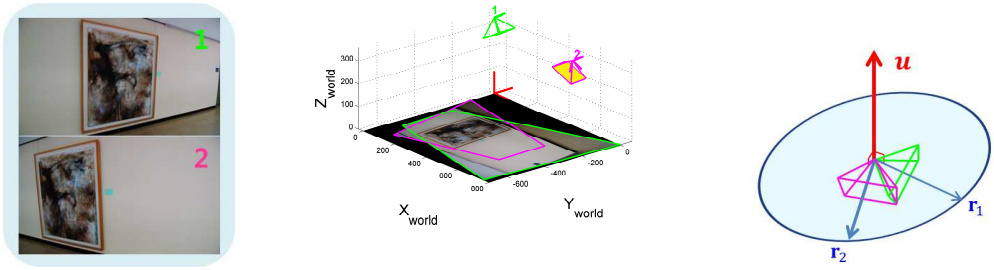
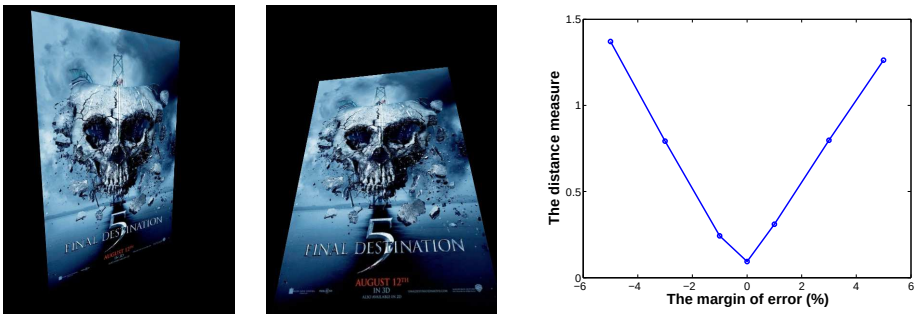


Figure 4.4: Illustration of notations in this paper.



(a) (b) (c)

Figure 4.5: Up-vector computation for skew removal in composites. (a) Two input images, (b) Visualization of estimated parameters, (c) Illustration of a up-vector $\mathbf{u}$.



(a) (b) (c)

Figure 4.6: (a), (b) Synthesized image pair, (c) Average value of (4.32) for the 275 pairs of synthesized images according to the margin of error.

(a)

(b) (c)

Figure 4.7: Comparison of the proposed result with the conventional image stitching method. (a) Six input images, (b) Result of *Autostitch*, (c) Result of the proposed method.

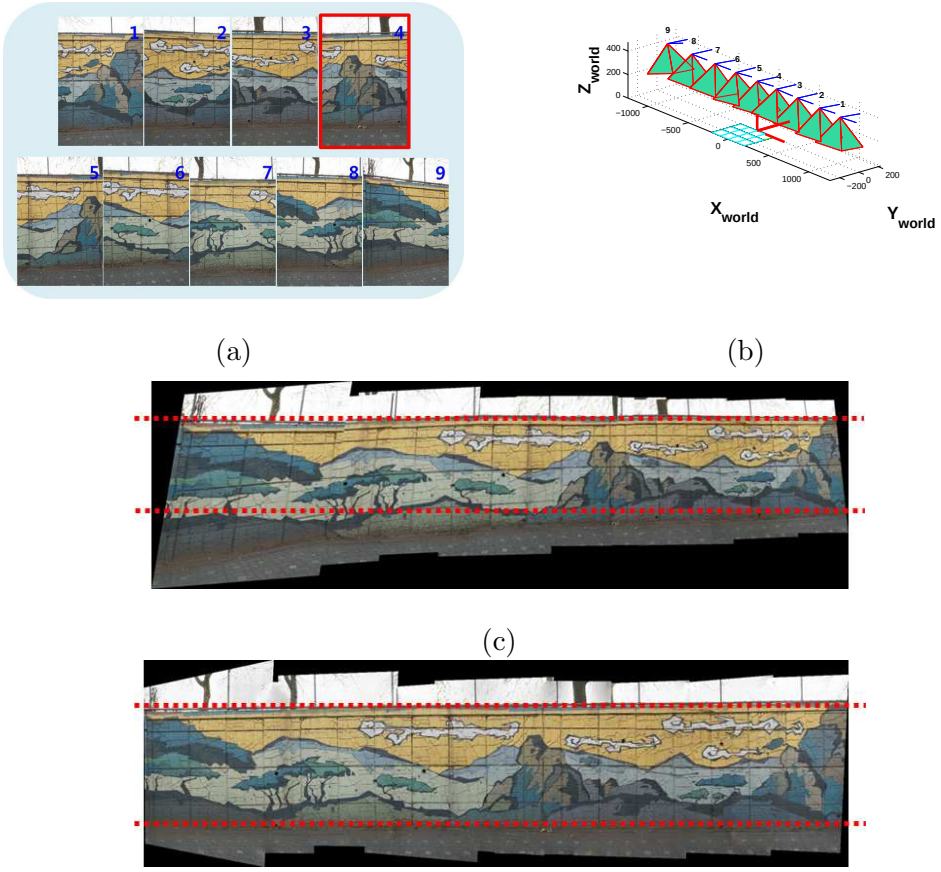
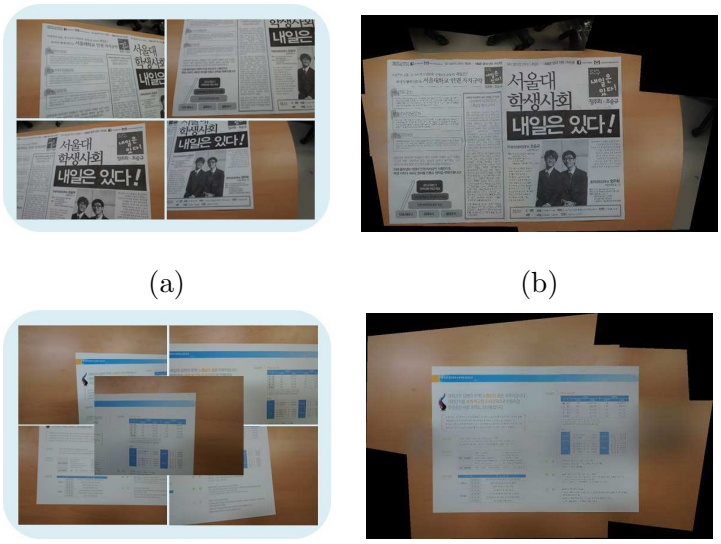


Figure 4.8: Image stitching result on the images captured by a moving camera. (a) Nine input images, (b) Estimated camera poses, (c) Image stitching result that considers the fourth image as a reference, (d) Image stitching result using the proposed method.



(a) (b)

(c) (d)

Figure 4.9: Image stitching results for document images. (a), (c) Input images, (b), (d) Image stitching results of the proposed method.

Chapter 5

Scene text detection and rectification

5.1 Introduction

5.1.1 Contribution

In general, most connected components (CCs) based scene text detection algorithms use prior knowledge on language model in grouping text candidates [18, 20, 19]. For example, the method in [19] assumes Latin languages whose characters consist of a single CC, therefore it considers only pairwise relation between horizontally adjacent CCs and is unconcerned about CCs in vertical direction for clustering text candidates. However, this language model can not be adopted in Asian characters such as Korean, Chinese, and Japanese since a single character in these languages consists several CCs including CCs along vertical direction. For this, new clustering method of detected text is proposed. In case of Asian characters, it is ambiguous which text line a CC belong to since a single character consists several CCs and

(a) (b)

Figure 5.1: Examples of Asian Text. In case of Asian character, it is hard to divide CCs into each text line without language model

some CCs in a text line can be closer to CCs in another text line than its original text line as shown in Fig. 5.1.

Hence hard decision clustering method [19] may classify CCs in a text-line into several groups or merge several text-lines into a group. To prevent this, soft decision clustering based on pairwise similarity between CCs is proposed. The proposed method computes similarity between all pairs of CCs and groups CCs sharing similar properties into a text-line. The main difference from the previous work is that the classification result is not fixed and is refined in the verification step. All the clusters are verified whether two or more text-lines are merged into a group. When the classification result is not correct, the clustering is performed more coarsely to divide the merged text-lines. This verification step reduces misclassification in the Asian characters significantly and prevents the degradation arisen when considering vertical relation between CCs in the Latin characters. However, the proposed method may generate more noisy clusters which consist of non-text CCs only due to the consideration of the wider range of correlation. For removing the noisy clusters,

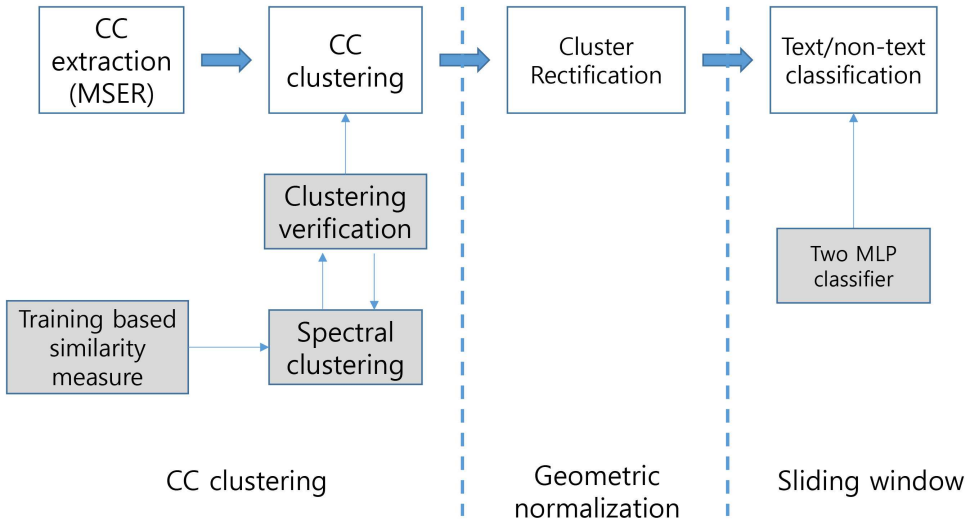


Figure 5.2: (The flowchart of the proposed method.)

more precise text/non-text classification is also proposed. The proposed algorithm shows better recall and precision in the Asian characters and also achieves competitive performance compared to the state of the art in the Latin characters.

5.1.2 Proposed approach

The proposed method use the maximally stable extremal region (MSER) algorithm [21] in Y-CB-CR color space to extract text candidates as the method in [19]. The detected CCs are grouped into text-lines using the proposed clustering method. In the cluster rectification step, detected text boxes are geometrically normalized by removing perspective distortion in the text. As the last step, the rectified text boxe are classified into text or non-text by employing two multi-layer perceptorn (MLP) classifiers. The details of each step is explained in following sections. The flow chart of the proposed method is shown in the Fig. 5.2.

The rest of the chapter is organized as follows. In Sec. 4.2, the proposed sim-

ilarity measure between pair of CCs and it clustering method are presented. After the clustering, the rectification algorithms of detected text box will be explained in Sec. 4.3. Experimental results on the Asian and the Latin characters are shown in Sec. 4.4.

5.2 Candidate region detection

5.2.1 CC extraction

In natural scene, text has various appearance in color, font, orientation, stroke width as shown in Fig. 5.1. In this case, general text-line extraction methods frequently fails to work due to difficulty in binarization [67, 68, 3, 69]. Fig. 5.5 shows the binarization result of a normal billboard image with scene text. Due to diversity in color in text and background, there are little text remained in the binarized image in Fig. 5.5-(b). To avoid such difficulty, the proposed method adopt the MSER algorithm [21] for detecting the text components. While text in natural scene has great diversity in appearance, there is a common property: small homogeneous regions surrounded by background pixel of separated intensity. The MSER algorithm is adequate to exploit this property as shown in the Fig. 5.5-(c). The proposed method extracted the MSER in each channel of the Y-Cb-Cr color space and repeats in the inverted image to detect brighter regions than their surroundings.

5.2.2 Computation of similarity between CCs

The proposed method groups the detected CCs into text-lines based on the pairwise similarity between CCs. Before clustering CCs, the similarity measure needs to be

(a) (b) (c)

Figure 5.3: An example of scene text image. (a) original image (b) binarization result (c) MSER result

defined. In the proposed method, when two CCs are parts of a single character or two adjacent characters in the same text-line, these CCs are called to be similar. The relations between CCs are described by 8-dimensional feature vector consisting of six geometrical feature, one stroke width related feature, and one color based feature. In computing geometric relation between CCs, width and height of bounding box of CCs, overlap length of two bounding boxes are employed as shown Fig. 5.4.

All the features are normalized with respect to scale of the bounding box :

$$\min\left(\frac{\max(h_i,h_j)}{\min(h_i,h_j)},\frac{\max(w_i,w_j)}{\min(w_i,w_j)}\right),\max\left(\frac{\max(h_i,h_j)}{\min(h_i,h_j)},\frac{\max(w_i,w_j)}{\min(w_i,w_j)}\right) \tag{5.1}$$

$$\max\left(\frac{ho_{ij}}{\min(w_i,w_j)},\frac{vo_{ij}}{\min(h_i,h_j)}\right),\frac{d_{ij}^{v}}{\min(h_i,h_j)}\times\frac{h_l}{w_l} \tag{5.2}$$

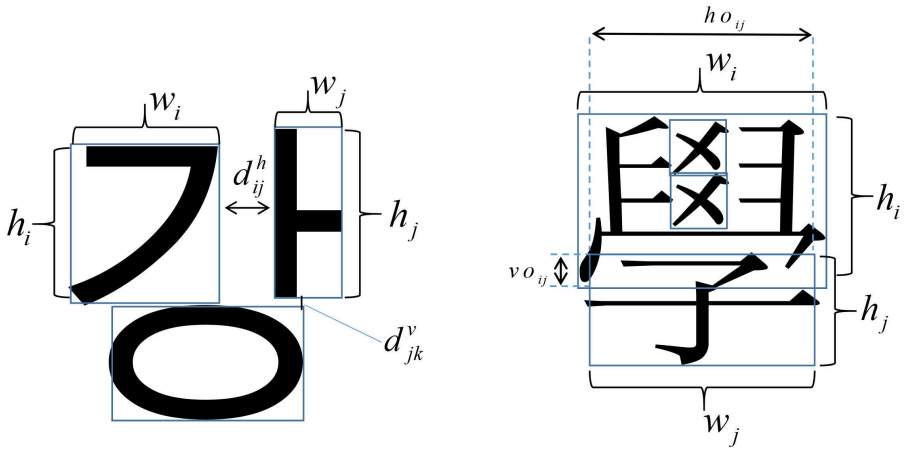


Figure 5.4: Illustration of geometrical relations between CCs

$$\max\left(\frac{|h_i - h_j|}{\min(h_i, h_j)} \times \frac{d^h_{ij}}{\min(w_i, w_j)}, \frac{|w_i - w_j|}{\min(w_i, w_j)} \times \frac{d^v_{ij}}{\min(h_i, h_j)}\right) \tag{5.3}$$

$$\frac{\max(s_i, s_j)}{\min(s_i, s_j)}, \frac{\max(sw_i, sw_j)}{\min(sw_i, sw_j)}. \tag{5.4}$$

To deal with the vertical relation between CCs equally with the horizontal relation, these two relations are bound to a single feature vector in (5.1), (5.2), and (5.3). In (5.2), h_l and w_l denote the height and the width of CC with lower center point than other CC, and this ratio of the height to the width penalizes the case where the lower CC has longer height than its width because that means the lower CC is like to come from other text-lines. Features in (5.2) penalize when the difference between the widths or the heights of two CCs is large especially when the distance between two CCs are far. In (5.4), s_i is number of pixels of i-th CC, and sw_i denotes the stroke width of i-th CC. The color based feature is defined as the euclidean distance between mean color vectors of two CCs in the R-G-B color space. By using the defined feature vectors, the gentle AdaBoost regressor [70, 71] is trained with the

training sample composed of English, Korean, Chinese, and Japanese. The resultant regressor outputs $[0, 1]$ by measuring the similarity between two CCs. This value is employed in the following clustering step.

5.2.3 CC clustering

The detected CCs are clustered into text-lines based on the pairwise similarity. Among the similarity based clustering method, spectral clustering is employed since it can subdivide the clusters as much as possible [72]. At fist, since the scale of similarity between text-lines is unknown, CCs are clustered in a coarse manner. The coarse clustering result usually includes clusters which contain more than one text-line since vertical relation between CCs are considered same as horizontal case. Therefore each cluster is verified whether it contains a single text-line or not based on the following criterion:

$$\Gamma = \min_{(i,j)\in\mathbb{N}_v} D(C_i, C_j), \tag{5.5}$$

where $\mathbb{N}_v$ is the set of pairs of CCs that have the horizontal overlap between bounding boxes as shown in the red rectangle of Fig. **??**-(c), $D(\cdot)$ is a function which measures similarity between two CCs C_i and C_j. When $\Gamma < \tau$, the cluster is subdivided into two text-lines. This process is repeated until all clusters pass the verification step.

5.3 Rectification of candidate region

Before classifying the detected clusters into text or non-text, all the clusters are rectified. In rectification, projective transformation is employed, which can remove

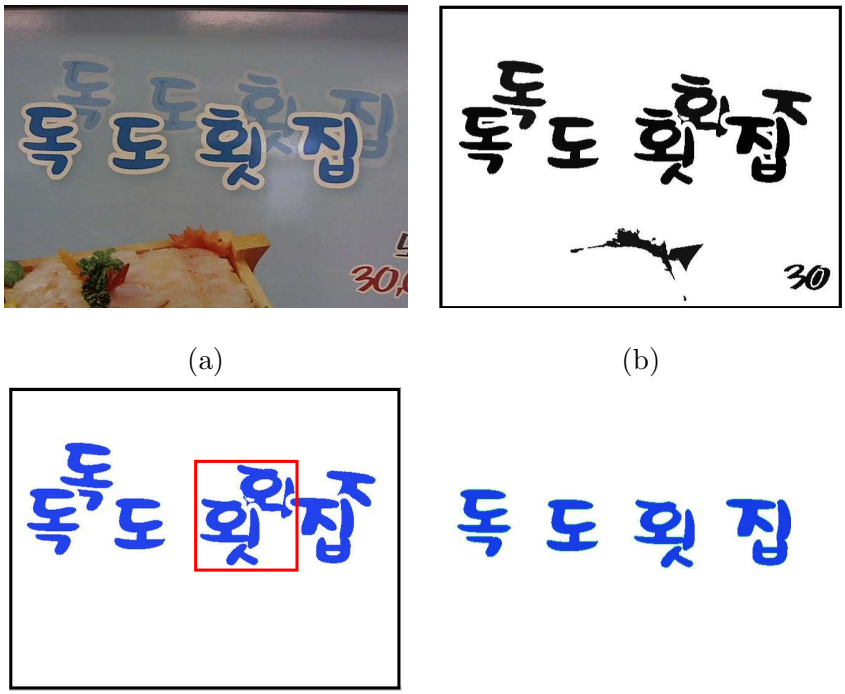


Figure 5.5: Result of the proposed clustering method. (a) Original image (b) MSER result (c) Coarse clustering result (d) Refined clustering result.

rotation, skew, perspective distortion remaining in the detected text. When the detected text boxes are assumed to be parallelograms, four boundaries of each box have to be estimated to rectify them [13]. Before estimating top and bottom boundaries, bounding boxes of CCs with large horizontal overlap are merged as shown in Fig. 5.6-(a) to reduce the effect of the various heights of bounding boxes in the Asian characters. Then, two sets of center points on the top and bottom lines of bounding boxes are built. On each set, histogram of orientation between all the pair of the points is computed, and the orientation of maximum bin is selected as a slope of the boundary line. In case of left and right boundaries, it is impossible to adopt this orientation histogram based method due to lack of samples. To address the problem, stroke based orientation estimation method which was proposed in the document rectification area is adopted [13]. Since many languages contain characters with vertical stroke, left and right boundaries are estimated by locating the vertical stroke. The detected text area is divided into two regions (left and right), then sub-region R containing vertical stroke is located on each region by optimizing the cost function :

$$J(R) = \sum_{x \in R} m(x) \left(\theta(x) - \bar{\theta}_R\right)^2 - \beta \sum_{x \in R} m(x), \tag{5.6}$$

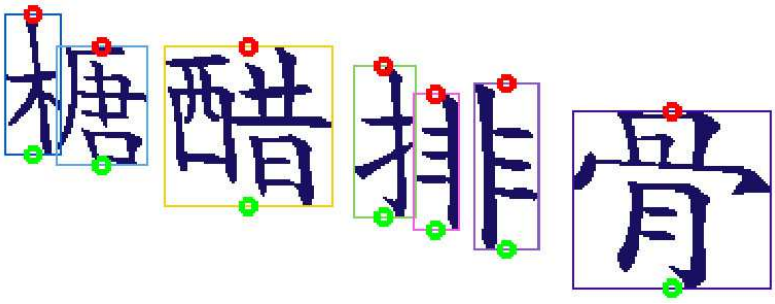
where $\theta(x)$ and $m(x)$ are the orientation and the magnitude of the gradient at a point x, $\bar{\theta}_R = \frac{\sum_{x \in R} m(x)\theta(x)}{\sum_{x \in R} m(x)}$ is a weighted average of gradient orientation in R. The first term in (5.6) is employed to measure the consistency of the gradient orientation in R, so that this value will be large when the weighted variance of the gradient orientation in R is large. The second term in (5.6) is employed to keep off the trivial solution $R = \emptyset$. This cost function is solved by the method proposed in [13], and

then both side boundaries can be estimated as shown in Fig. 5.6-(b). Using the estimated boundaries, the rectified image is obtained as shown in Fig. 5.6-(c).

5.4 Text/non-text classification

The rectified text is classified into text or non-text by two MLP classifiers [73]. The proposed method adopts text/non-text classifier in the method in [19] as the first classifier. The method in [19] splits the rectified text image into overlapping squares and classifies each sub-region into text or non-text. After that, results of the square images are accumulated to classify original block. However, since this method does not consider pairwise relation between square images, repetitive patterns from man-made structures such as building facades and windows are frequently mis-classified as shown in Fig. 5.7. To address the problem, the proposed method employs the MLP which considers pairwise relations between adjacent square images.

The differences between two feature vectors are used as new features for the second MLP classifier as shown in Fig. 5.8. The second MLP is trained using the same training samples as the first MLP, and also has same structures as the first structures. The detected text region is classified as text when both classifier classify the region as text. If one of the classifier classifies the region into non-text, the detected text region is rejected.

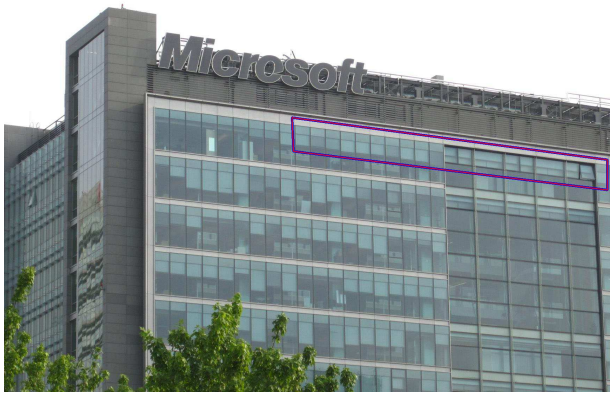


(a)

(b)

(c)

Figure 5.6: Rectification of the detected text box. (a) Merged bounding boxes of CCs. Red circles is the center points of top lines, and green circles is the center points of bottom lines (b) Estimated boundaries of text box (c) Rectified images using the estimated boundaries.



(a) Detected text box

(b) Rectified text box

(c) Detected text box

(d) Rectified text box

Figure 5.7: Examples of mis-classification with a single MLP classifier. The detected text boxes are displayed as red rectangles.

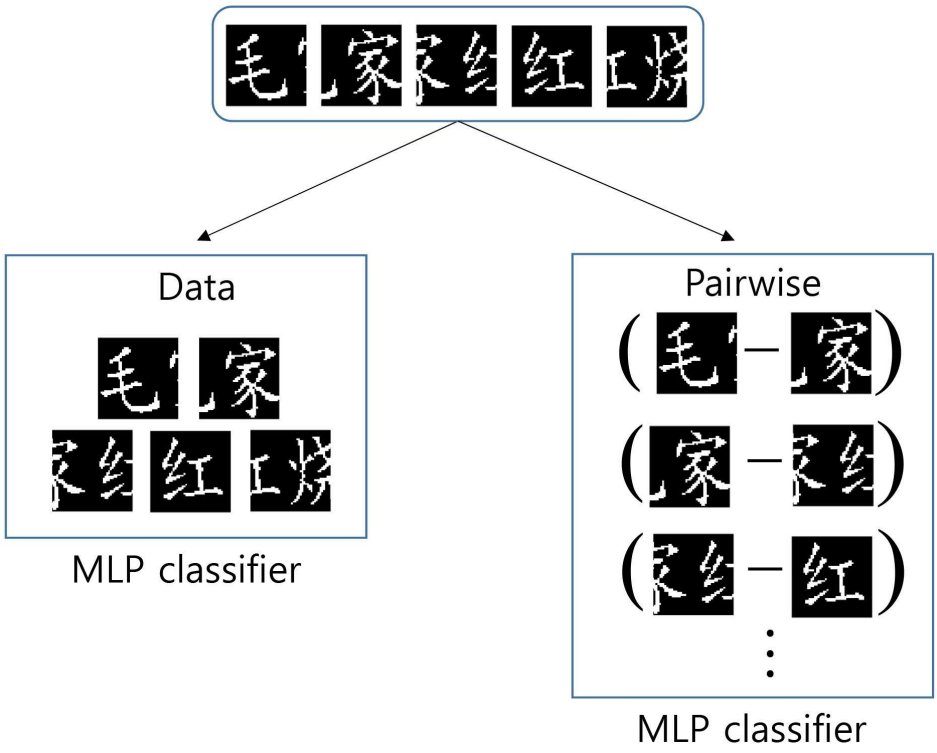


Figure 5.8: Proposed two MLP classifiers.

Table 5.1: Evaluation on ICDAR 2011 dataset

Algorithm	Precision	Recall	f-measure
Yao's [18]	0.69	0.66	0.67
Neumann's [20]	0.793	0.664	0.723
Koo's [19]	0.8144	**0.6868**	0.7452
Proposed method	**0.8464**	0.6681	**0.7468**

5.5 Experimental result

5.5.1 Experimental results on ICDAR 2011 dataset

At first, the proposed method is evaluated on ICDAR 2011 dataset [74], which is the standard English data set for scene text detection.

Table. 5.1 shows the precision, recall, and f-measure of the state-of-the-art algorithms. Proposed method shows slightly worse recall rate than Koo's method because the proposed method doesn't employ the assumption of English language model unlike Koo's method. In terms of precision, the proposed method shows best performance due to the proposed rectification step and the text/non-text classifier using two MLP. As a result, the proposed method obtains best f-measure without the assumption that the characters to be detected are English.

5.5.2 Experimental results on the Asian character dataset

For evaluating the performance on the Asian characters, the Asian character data set which contains 200 images consisting of Korean, Chinese, and Japanese. The proposed method and the state-of-the-art algorithm in English characters are tested on this data set. The results are shown in Table. 5.2. The proposed method shows

Table 5.2: Evaluation on Asian character dataset

Algorithm	Precision	Recall	f-measure
Koo's [19]	0.2	0.16	0.1778
Proposed method	**0.34**	**0.26**	**0.29**

better results than Koo's method in terms of precision and recall on the Asian character data set. The examples of the Asian character data set and the results of the proposed method are shown in Fig. 5.9.

(a) (b)

(c) (d)

Figure 5.9: Results of the proposed method in the Asian character data set. Red rectangles are the detected text box.

Chapter 6

Conclusion

In this dissertation, the rectification methods for document images or scene text are proposed, based on text-line, feature point, and text detection. Specifically, two rectification methods via optimization of the cost functions are proposed, and a scene text detection algorithm with the text rectification is developed. The most suitable rectification algorithm among the proposed methods can be selected depending on the kinds of document, layout, and sparsity of text.

Firstly a new framework for the rectification of documents with dense text has been proposed. The contributions in this work are summarized as: (i) The dewarping problem is formulated as an optimization problem based on the given text-line and text-block information. While most conventional algorithms consist of a series of steps where each step assumes that the previous steps are successful, the proposed method addresses the same problem by optimizing a single cost function. (ii) The proposed method uses discrete point set on the text-lines, which can reduce the effect of the noise in the text-lines and the error occurred in continuous text-line fitting. Moreover, the discrete text-line model facilitates the optimization of the

proposed cost function. (iii) The proposed method does not need special assumption on document layout (such as single column format or justified page layout). Rather, the proposed method detects text-blocks and imposes certain constraints according to their properties, which gives the robustness to the variation of layouts and background. (iv) The proposed method deals with not only planar documents but also curved surfaces in the same framework, which has been treated in a different manner in most previous works. Moreover, the proposed method can be extended to the unfolded book surfaces by changing the surface models. Extensive experiments demonstrate the robustness of proposed method to camera poses and surface complexities, and its results compare favorably with the conventional methods on the standard dataset. The proposed method assumes that the shape of document can be modeled by the general cylindrical surface (GCS). If the shape of surface violates this assumption, the algorithm does not work. Hence, one possible topic for future work is an extension of the GCS model to more general model. For example, employing the developable surface model can reduce error when the document surface is distorted by the complex deformation. However, more sophisticated optimization techniques need to be developed.

Secondly a unified approach to the image stitching and rectification problem for the large document or documents with little text has been proposed. While the conventional methods have been developed independently either for (i) fixed-optical-center case or (ii) moving camera capturing plane-target case, the proposed framework is able to handle both cases in the same framework. For this, the six parameters of each camera is estimated by optimizing the proposed cost function. Since the cost function is defined as a sum of squared error between corresponding points on the reference plane where the documents is laid on, it can be minimized

via the Levenberg-Marquardt algorithm. From the estimated camera poses, the proposed method performs metric rectification when there are enough camera motions (or differences), otherwise, it yields stitching results without rectification like the conventional methods. In quantitative evaluation of the proposed method, registration error are less than 1.5 pixels for a range of deformations, and it also has little error in the classification of motion models. In experiments on real images, the proposed method shows satisfactory results compared with conventional methods. In addition to document stitching and rectification, the proposed method can detect a plane and estimate its pose using multiple images, and these can be applied to augmented reality (AR) application.

Lastly a new approach to the rectification of text in the natural scene is proposed. For the rectification of scene text, a new framework for detecting the scene text is also proposed. Previous scene text detection algorithms assume certain language model, and frequently fail for the characters from other languages. On the other hand, the proposed method is robust to the language models and works for the Asian characters as well as Alphabets from English. To make the algorithm insensitive to language model, vertical relations between text components are dealt with the same manner as the horizontal relations. However, since this approach may merge several text-lines into a group, a new clustering method employing coarse-to-fine strategy is proposed. To be specific, the proposed method clusters the candidate regions in terms of adjacency between the detected regions, and verifies the resultant clusters in a coarse-to-fine manner. The adjacency measure is trained on the data set labeled with the bounding box of text region. For filtering the non-text clusters, a new classifier employing pairwise relations between the subregions of a detected text is proposed. To improve the accuracy of the classifier, the detected text region is

rectified before filtering by optimization-based vertical stroke estimation method. In the experiments, the proposed method achieves state-of-the-art performance in the Asian characters as well as in the standard English dataset. In addition to the Asian characters, the proposed method is expected to deal with a variety of characters if the sufficient datasets for training classifiers are available.

Bibliography

[1] L. O'Gorman, "The document spectrum for page layout analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1162–1173, 1993.

[2] G. Nagy, "Twenty years of document image analysis in pami," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 38–62, 2000.

[3] H. I. Koo and N. I. Cho, "State estimation in a document image and its application in text block identification and text line extraction," in *Proc. 11th European Conf. Computer vision*, pp. 421–434, 2010.

[4] G. Meng, C. Pan, S. Xiang, J. Duan, and N. Zheng, "Metric rectification of curved document images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 707–722, 2012.

[5] Z. Zhang and C. Tan, "Correcting document image warping based on regression of curved text lines," in *Proc. 7th Int'l Conf. Document Analysis and Recognition*, vol. 1, pp. 589–593, 2003.

[6] A. Ulges, C. H. Lampert, and T. Breuel, "Document image dewarping using robust estimation of curled text lines," in *Proc. 8th Int'l Conf. Document Analysis and Recognition*, vol. 2, pp. 1001–1005, 2005.

[7] M. Brown, M. Sun, R. Yang, L. Yun, and W. Seales, "Restoring 2d content from distorted documents," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 11, pp. 1904–1916, 2007.

[8] L. Zhang, Y. Zhang, and C. Tan, "An improved physically-based method for geometric restoration of distorted document images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 4, pp. 728–734, 2008.

[9] F. Courteille, A. Crouzil, J.-D. Durou, and P. Gurdjos, "Shape from shading for the digitization of curved documents," *Machine Vision and Applications*, vol. 18, no. 5, pp. 301–316, 2007.

[10] L. Zhang, A. M. Yip, M. S. Brown, and C. L. Tan, "A unified framework for document restoration using inpainting and shape-from-shading," *Pattern Recognition*, vol. 42, no. 11, pp. 2961–2978, 2009.

[11] B. Fu, M. Wu, R. Li, W. Li, Z. Xu, and C. Yang, "A model-based book dewarping method using text line detection," in *Proc. 2nd Int. Workshop on Camera Based Document Analysis and Recognition*, pp. 63–70, Sep. 2007.

[12] N. Stamatopoulos, B. Gatos, I. Pratikakis, and S. Perantonis, "Goal-oriented rectification of camera-based document images," *IEEE Trans. Image Process.*, vol. 20, no. 4, pp. 910–920, 2011.

[13] Y. Tian and S. Narasimhan, "Rectification and 3d reconstruction of curved document images," in *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, pp. 377–384, 2011.

[14] J. Liang, D. DeMenthon, and D. Doermann, "Geometric rectification of camera-captured document images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 4, pp. 591–605, 2008.

[15] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[16] R. Szeliski, "Image alignment and stitching: A tutorial," *Found. Trends. Comput. Graph. Vis.*, vol. 2, no. 1, pp. 1–104, Jan. 2006.

[17] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *Int. J. Comput. Vision*, vol. 74, no. 1, pp. 59–73, Aug. 2007.

[18] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, pp. 1083–1090, 2012.

[19] H. I. Koo and D. H. Kim, "Scene text detection via connected component clustering and nontext filtering," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2296–2305, 2013.

[20] L. Neumann and J. Matas, "Scene text localization and recognition with oriented stroke detection," in *Proc. IEEE Int'l Conf. Computer Vision*, 2013.

[21] D. Nistér and H. Stewénius, "Linear time maximally stable extremal regions," in *Proc. 10th European Conf. Computer Vision: Part II*, ser. ECCV '08, pp. 183–196. Berlin, Heidelberg: Springer-Verlag, 2008.

[22] K. Jung, "Text information extraction in images and video: a survey," *Pattern Recognition*, vol. 37, no. 5, pp. 977–997, May 2004.

[23] J. Liang, D. Doermann, and H. Li, "Camera-based analysis of text and documents: a survey," *Int'l J. Document Analysis and Recognition*, vol. 7, no. 2, pp. 84–104–104, Jul. 2005.

[24] J. Zhang and R. Kasturi, "Extraction of text objects in video documents: Recent progress," in *Proc. 8th Int'l Workshop Document Analysis Systems*, pp. 5–17, Sep. 2008.

[25] Y.-C. Tsoi and M. Brown, "Multi-view document rectification using boundary," in *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[26] H. I. Koo, J. Kim, and N. I. Cho, "Composition of a dewarped and enhanced document image from two view images," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1551–1562, 2009.

[27] Z. Zhang, X. Liang, and Y. Ma, "Unwrapping low-rank textures on generalized cylindrical surfaces," in *Proc. 13th IEEE Intl Conf. Computer Vision*, pp. 1347–1354, 2011.

[28] Y.-C. Tsoi and M. Brown, "Geometric and shading correction for images of printed materials: a unified approach using boundary," in *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 240–246, 2004.

[29] A. Zappal, A. Gee, and M. Taylor, "Document mosaicing," *Image and Vision Computing*, vol. 17, no. 8, pp. 589 – 595, 1999.

[30] J. Liang, D. DeMenthon, and D. Doermann, "Camera-based document image mosaicing," in *Proc. 18th Int'l Conf. Pattern Recognition*, ser. ICPR '06, vol. 2, pp. 476–479. Washington, DC, USA: IEEE Computer Society, 2006.

[31] ——, "Mosaicing of camera-captured document images," *Computer Vision and Image Understanding*, vol. 113, no. 4, pp. 572 – 579, 2009.

[32] M. Ligang and Y. Yongjuan, "Automatic document image mosaicing algorithm with hand-held camera," in *Proc. 2nd Int'l Conf. Intelligent Control and Information Processing*, vol. 2, pp. 1094–1097, Jul. 2011.

[33] P. Shivakumara, G. H. Kumar, D. Guru, and P. Nagabhushan, "Sliding window based approach for document image mosaicing," *Image and Vision Computing*, vol. 24, no. 1, pp. 94 – 100, 2006.

[34] A. Iketani, T. Sato, S. Ikeda, M. Kanbara, N. Nakajima, and N. Yokoya, "Super-resolved video mosaicing for documents based on extrinsic camera parameter estimation." in *Proc. 7th Asian Conf. Computer Vision*, ser. Lecture Notes in Computer Science, vol. 3852, pp. 101–110. Springer, 2006.

[35] ——, "Video mosaicing based on structure from motion for distortion-free document digitization." in *Proc. 8th Asian Conf. Computer Vision*, ser. Lecture Notes in Computer Science, vol. 4844, pp. 73–84. Springer, 2007.

[36] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," in *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, ser. CVPR'04, pp. 366–373. Washington, DC, USA: IEEE Computer Society, 2004.

[37] X. Chen and A. Yuille, "A time-efficient cascade for real-time object detection: With applications for the visually impaired," in *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition Workshops*, pp. 28–28, June 2005.

[38] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004.

[39] J.-J. Lee, P.-H. Lee, S.-W. Lee, A. Yuille, and C. Koch, "Adaboost for text detection in natural scene," in *Proc. 11th Int'l Conf. Document Analysis and Recognition*, pp. 429–434, 2011.

[40] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, pp. 2963–2970, 2010.

[41] H. Chen, S. Tsai, G. Schroth, D. Chen, R. Grzeszczuk, and B. Girod, "Robust text detection in natural images with edge-enhanced maximally stable extremal regions," in *Proc. 18th IEEE Int'l Conf. Image Processing*, pp. 2609–2612, 2011.

[42] L. Neumann and J. Matas, "Text localization in real-world images using efficiently pruned exhaustive search," in *Proc. 11th Int'l Conf. Document Analysis and Recognition*, pp. 687–691, 2011.

[43] H. Cao, X. Ding, and C. Liu, "A cylindrical surface model to rectify the bound document image," in *Proc. 9th IEEE Int'l Conf. Computer Vision*, vol. 1, pp. 228–233, 2003.

[44] M. Brown and Y.-C. Tsoi, "Geometric and shading correction for images of printed materials using boundary," *IEEE Trans. Image Process.*, vol. 15, no. 6, pp. 1544–1554, 2006.

[45] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C : The Art of Scientific Computing.* Cambridge University Press, 1992.

[46] M. Lourakis, "levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++," [web page] `http://www.ics.forth.gr/~lourakis/levmar/`, Jul. 2004, [Accessed on 31 Jan. 2005.].

[47] F. Shafait, "Document image dewarping contest," in *Proc. 2nd Int. Workshop on Camera-Based Document Analysis and Recognition*, pp. 181–188, 2007.

[48] A. Masalovitch and L. Mestetskiy, "Usage of continuous skeletal image representation for document images dewarping," in *Proc. 2nd Int. Workshop on Camera-Based Document Analysis and Recognition*, Curitiba, Brazil, Sep. 2007.

[49] B. Gatos, I. Pratikakis, and K. Ntirogiannis, "Segmentation based recovery of arbitrarily warped document images," in *Proc. 9th Int'l Conf. Document Analysis and Recognition*, vol. 2, pp. 989–993, 2007.

[50] S. S. Bukhari, F. Shafait, and T. M. Breuel, "Dewarping of document images using coupled-snakes," in *Proc. 3rd Int. Workshop on Camera-Based Document Analysis and Recognition*, Barcelona, Spain, Jul. 2009.

[51] M. Frigge, D. C. Hoaglin, and B. Iglewicz, "Some implementations of the boxplot," *The American Statistician*, vol. 43, no. 1, pp. 50–54, 1989.

[52] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, pp. 707–710, 1966.

[53] P. Clark and M. Mirmehdi, "Estimating the orientation and recovery of text planes in a single image," in *Proc. 13th British Machine Vision Conference*, pp. 421–430. BMVA Press, 2001.

[54] M. Pilu, "Extraction of illusory linear clues in perspectively skewed documents," in *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. I–363–I–368, 2001.

[55] L.-Y. Duan, R. Ji, Z. Chen, T. Huang, and W. Gao, "Towards mobile document image retrieval for digital library," *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 346–359, Feb. 2014.

[56] W. Lee, Y. Pack, and V. Lepetit, "Video-based *In Situ* tagging on mobile phones," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, pp. 1487–1496, 2011.

[57] C. Xu, B. Kuipers, and A. Murarka, "3d pose estimation for planes," in *Proc. 12th IEEE Int'l Conf. Computer Vision Workshops*, pp. 673–680, Oct. 2009.

[58] J. Mor, "The levenberg-marquardt algorithm: Implementation and theory," in *Numerical Analysis*, ser. Lecture Notes in Mathematics, G. Watson, Ed. Springer Berlin Heidelberg, 1978, vol. 630, pp. 105–116.

[59] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[60] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[61] H. I. Koo, B. S. Kim, and N. I. Cho, "A new method to find an optimal warping function in image stitching," in *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, Apr. 2009.

[62] B. S. Kim, H. I. Koo, and N.-I. Cho, "A new image projection method for panoramic image stitching," in *Proc. IEEE Int'l Workshop Multimedia Signal Processing*, pp. 128–132, Oct. 2010.

[63] [Online]. Available: http://www.autostitch.net/

[64] P. J. Burt, Edward, and E. H. Adelson, "The laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. 31, pp. 532–540, 1983.

[65] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, pp. 376–380, 1991.

[66] J. Hannuksela, P. Sangi, J. Heikkil, X. Liu, and D. S. Doermann, "Document image mosaicing with mobile phones." in *Proc. 14th Int'l Conf. Image Analysis and Processing*, R. Cucchiara, Ed., pp. 575–582. IEEE Computer Society, 2007.

[67] S. Basu, C. Chaudhuri, M. Kundu, M. Nasipuri, and D. Basu, "Text line extraction from multi-skewed handwritten documents," *Pattern Recognition*, vol. 40, no. 6, pp. 1825 – 1839, 2007.

[68] G. Louloudis, B. Gatos, I. Pratikakis, and C. Halatsis, "Text line and word segmentation of handwritten documents," *Pattern Recognition*, vol. 42, no. 12, pp. 3169 – 3183, 2009.

[69] H. I. Koo and N.-I. Cho, "Text-line extraction in handwritten chinese documents based on an energy minimization framework," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 1169–1175, Mar. 2012.

[70] J. Friedman, T. Hastie, and R. Tibshirani, "Additive Logistic Regression: a Statistical View of Boosting," *The Annals of Statistics*, vol. 28, no. 2, 1998.

[71] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. Cambridge, MA: O'Reilly, 2008.

[72] U. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, Dec. 2007.

[73] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.

[74] A. Shahab, F. Shafait, and A. Dengel, "Icdar 2011 robust reading competition challenge 2: Reading text in scene images." in *Proc. 11th Int'l Conf. Document Analysis and Recognition*, pp. 1491–1496, 2011.

초록

문서나 텍스틀 카메라로 촬영한 후 이 영상을 활용한 문서 검색, 책 읽어주기, 텍스트 기반의 증강 현실과 같은 어플리케이션에서는 영상에서 텍스트를 검출 및 인식하기 위하여 텍스트가 적힌 평면의 원래 모습(texture)을 아는 것이 중요하다. 하지만 카메라로 촬영한 대부분의 영상에서 텍스트가 적힌 평면은 perspective 왜곡과 책과 같이 종이의 구부러짐으로 인하여 원래 모습과는 다른 모습을 보인다. 이렇게 카메라로 촬영한 왜곡된 문서 영상에서 텍스트가 적힌 표면의 원래 모습 또는 텍스트의 원래 모습을 복원하는 과정을 문서 평활화라고 한다. 본 논문에서는 텍스트 인식률과 시각적 품질을 높이기 위한 새로운 문서 평활화 방법을 제안한다. 제안하는 방법은 문서의 형태에 따라 세가지 타입으로 분류되며 각각의 알고리즘이 기여하는 부분은 다음과 같다.

첫 번째 알고리즘은 책과 같이 텍스트 라인의 수가 많고 명확하게 검출되는 문서에 활용 가능한 문서 평활화 방법이다. 기존의 방법은

텍스트 라인의 곡선을 그대로 이용하는 반면에 제안하는 방법은 텍스트 라인을 구성하는 각각의 텍스트를 하나의 점으로 보고 이들의 불연속 집합을 활용하여 문서 평활화를 최적화 문제로 모델링한다. 이때 종이의 구부러짐은 generalized cylindrical surface (GCS)로 근사하고 perspective 왜곡은 카메라의 회전으로 근사하여 모델링한다. 이 모델과 텍스트의 불연속 집합을 가지고 새로운 목적 함수를 개발하였으며, 이 목적 함수의 최적화를 통하여 모델의 파라미터를 구하게 된다. 제안하는 목적 함수는 문단의 정렬 형태, 줄 간격, 텍스트 라인의 수평성과 같이 문서의 일반적인 특징을 활용하여 설계된다. 텍스트의 불연속 집합을 사용하기 때문에 제안하는 목적 함수는 쉽게 최적화할 수 있는 형태를 가지게 되며 Levenberg-Marquadt 알고리즘으로 최적화하게 된다. 실험 결과 제안하는 방법은 다양한 레이아웃과 구부러진 형태의 문서에서 안정적으로 동작하였고, 표준 데이터세트에서 기존 알고리즘에 비해 더 높은 정확도를 보였다.

두 번째 알고리즘은 벽화 또는 현수막과 같이 한 장의 영상에 전체를 담기 힘든 커다란 문서를 여러 장으로 나누어 촬영한 후 영상의 합성과 문서 평활화를 동시에 수행하는 방법이다. 이 방법은 일반적인 영상 합성과 비슷해 보이지만 가정하고 있는 카메라 모션에서 차이를 보인다. 일반적인 영상 합성 방법은 카메라가 한 곳에 고정된 상태에서 회전하여 촬영한 영상을 가정하지만 문서를 촬영할 경우에는 문서가 고정되어 있고 카메라를 움직여 촬영하게 된다. 본 논문에서는 이러한 영상들 사이에 존재하는 카메라의 움직임을 파라미터를 사용하여 새롭게 모델링하였으며, 문서가 위치한 평면에 제약조건을 부여하는 새로운 목적 함수를 제안한다. 제안하는 목적 함수의 최적화를 통하여 부가적인 정보 없이 영상만을 이용하여 문서 평면의 물리적으로 올바른 위치를 추정할 수 있으며, 추정한 문서 평면의 위치를 이용하여 정확한 문서 평활화를 수행할 수 있게 된다. 제안하는

방법은 텍스트를 활용하지 않기 때문에 문서뿐만 아니라 건물의 표면 등 다양한 평면에 적용 가능하다.

세 번째 알고리즘은 표지판과 같이 획이 명확하게 구분되는 일상적인 텍스트에 적용 가능한 문서 평활화 방법이다. 이를 위하여 언어의 형태에 강인한 새로운 텍스트 검출 방법을 제안하며, 제안하는 방법을 활용하여 텍스트를 평활화하게 된다. 기존의 텍스트 검출 방법은 영어 알파벳과 같이 하나의 텍스트가 하나의 connected component (CC)로 이루어져 있다고 가정하기 때문에 한글, 한자, 일본어와 같이 이러한 가정이 성립하지 않은 글자의 경우에는 제대로 동작하지 않은 단점이 있다. 이를 해결하기 위하여 제안하는 방법은 텍스트 후보 영역(CC)을 검출한 후 이 들 사이의 유사성에 기반한 클러스트링 방법으로 CC를 그룹화한다. 이때 한 글자 안에서 존재하는 CC 사이의 다양한 관계를 모델링하기 위하여, 여러 종류의 언어가 포함된 데이터 세트에서 CC 사이의 수평, 수직 방향의 겹친 정도 혹은 거리 등을 feature vector로 추출한 후 이를 활용하여 유사성을 측정하는 기준을 training한다. 클러스터링 후 잘못 검출된 텍스트 영역은 비텍스트 검출기를 이용하여 제거한 후, 최종적으로 남은 텍스트 영역들은 위치와 방향을 고려하여 텍스트 라인으로 분리되거나 합쳐진다. 검출된 텍스트는 텍스트 라인의 방향과 수직 방향의 획을 이용하여 평활화된다. 실험 결과 제안하는 방법은 아시아 언어뿐만 아니라 영어에서도 기존의 방법보다 나은 결과를 보였다.

주요어: 문서 영상, 문서 평활화, 문서 합성, 텍스트 검출

학번 : 2008-20836